



Universidad  
Carlos III de Madrid

Departamento de Tecnología Electrónica

PROYECTO FIN DE CARRERA

# **Diseño de una Placa de Desarrollo para microcontrolador PIC18F con conectividad IEEE 802.15.4**

Autor: Raúl Figueruelo Carretero

Tutor: Marta Ruiz Llata

Leganés, Diciembre de 2011



# Agradecimientos

En primer lugar, agradecer a todos los que de forma directa me han ayudado a finalizar con éxito el Proyecto Fin de Carrera. A Marta, por prestar su ayuda en todo momento, y al equipo de la Oficina Técnica de la UC3M, que han colaborado con el diseño de las pistas del circuito integrado y la organización de los componentes.

Tampoco quiero olvidar a mi familia y amigos, que cuando las cosas no iban bien, estaban ahí para apoyarme.



# Resumen

*Wireless* o sin cables. Ese es uno de los términos más utilizados para definir la tendencia de las comunicaciones. Y es que con el paso de los años, los usuarios cada vez demandan más la posibilidad de estar continuamente conectados sin la necesidad de estar enganchados a un cable. Las comunicaciones inalámbricas cuentan con grandes ventajas: flexibilidad, rapidez de instalación, bajos costes de infraestructuras, etc. La industria y los gobiernos ha establecido multitud de especificaciones y protocolos para regular el mercado, cubrir las necesidades de los usuarios y abrir nuevas oportunidades de negocio.

Con esta idea, se pretende diseñar y fabricar un sistema complejo basado en un microcontrolador que incorpore el uso de un protocolo de comunicaciones inalámbricas extendido en el mercado. Además, también se persigue incorporar otros elementos de entrada/salida para disponer de un sistema de desarrollo completo.

Con estos objetivos presentes, se estudian las diferentes especificaciones existentes para redes WPAN y WBAN catalogadas en la familia de estándares IEEE 802.15. Una vez discriminado el IEEE 802.15.4 como el más adecuado se ha elegido el protocolo propietario Microchip Wireless (MiWi).

Para fabricar el sistema, se ha desarrollado un prototipo hardware basado en un microcontrolador PIC18F2525, alrededor del cual se han conectado los dispositivos necesarios para utilizar las comunicaciones e interaccionar con el usuario: transceptor, *display* gráfico monocromo, pantalla táctil y comunicación serie RS232. Además, se ha otorgado a la placa de desarrollo cierta flexibilidad para que, en un futuro, puedan añadirse nuevos elementos.

Junto al hardware, se ha trabajado en el software para iniciar, controlar y utilizar todos los componentes incluidos. Este software sirve de demostración a la hora de elaborar un guión para diversas prácticas de laboratorio que pueden utilizar parte o la totalidad del prototipo aquí diseñado.

**Palabras clave:** Placa de desarrollo, IEEE 802.15.4, MiWi, Prácticas de Microprocesadores.



# Abstract

Wireless. This it is one of the most used terms to define the communications trend. And it is because of the years went by, users are demanding more possibilities of being constantly connected without needing of being hooked to a cable. Wireless communications has big benefits: flexibility, fast installation, low costs in infrastructures, etc. Industry and governments has established multitude of specifications and protocols to regulate the market, to serve users' needs and to open new business opportunities.

With this idea, it claims to design and make a complex system based on a microcontroller which incorporates an application protocol of wireless communications extended in to the market. Also, it is going after incorporate other input / output elements for having a complete development system.

With these present aims, the different existing specifications are studied for networks WPAN and WBAN catalogued in the family of standards IEEE 802.15. Once discriminated the IEEE 802.15.4 since the most suitable the proprietary protocol has chosen Microchip Wireless (MiWi).

To make the system, a prototype has developed hardware based on a microcontroller PIC18F2525, about whom the necessary devices have connected to use the communications and to interact with the user: transceiver, display, tactile screen and communication by RS232. In addition, certain flexibility has been granted to the development board. Also, certain flexibility has been granted to the development board in order that, in a future, new elements could be added.

Close to the hardware, one has been employed at the software to initiate, to control and to use all the included components. This software uses as demonstration at the moment of elaborating a script for diverse practices of laboratory that can use part or the totality of the prototype here designed.

**Key words:** Development Board, IEEE 802.15.4, MiWi Practices of microprocessors.





# Índice general

<b>Introducción</b>	<b>1</b>
1.1 Introducción	1
1.2 Objetivos del proyecto	2
1.3 Materiales utilizados	3
1.4 Fases de desarrollo	3
1.5 Contenido de la memoria	5
 <b>Redes WPAN</b>	 <b>7</b>
2.1 Introducción	7
2.2 IEEE 802.15.1	8
2.3 IEEE 802.15.2	9
2.4 IEEE 802.15.3	10
2.5 IEEE 802.15.4	11
2.6 IEEE 802.15.5	12
2.7 Otros estándares 802.15: 802.15.6 y 802.15.7	13
2.8 Comparativa y conclusiones	13
 <b>Diseño e implementación del prototipo</b>	 <b>15</b>
3.1 Diagrama de bloques	15
3.2 Microcontrolador PIC18F2525	16
3.3 Transceptor MRF24J40MA	18
3.4 Pantalla LCD	19
	 IX

3.5	Pantalla táctil	22
3.6	Comunicación RS232: ADM3202 y conector DB9	23
3.7	Regulador de tensión LM317	25
3.8	Prototipos	26
3.9	Conclusiones	28
<b>Protocolo Wireless</b>		<b>31</b>
4.1	Introducción	31
4.2	MiWi frente a Zigbee	32
4.3	Pila de Capas de MiWi	34
4.3.1	Capa de aplicación	34
4.3.2	Capa de red	35
4.3.3	Capa MAC	37
4.3.4	Capa física	39
4.3.5	Formato de mensaje	42
4.3.6	Seguridad	43
4.4	Mercado y aplicaciones	45
4.5	Diseño e implementación de MiWi en la placa de desarrollo	45
4.5.1	Nivel de aplicación	46
4.5.2	MiApp	47
4.5.3	Nivel de MiWi P2P	48
4.5.4	MiMAC	49
4.5.5	Nivel del Transceptor MRF24J40	50
4.6	Conclusiones	50
<b>Ejemplo de uso: Monitorización de Temperaturas</b>		<b>51</b>
5.1	Introducción	51
5.2	Incorporación del sensor LM71 al sistema HW	52
5.3	Descripción de las funcionalidades implementadas en los dispositivos	53
5.3.1	Coordinador	53
5.3.2	Procesado de paquetes recibidos por el coordinador	55
5.3.3	Nodo satélite RFD	56
5.4	Conclusiones	57
<b>Presupuesto</b>		<b>59</b>

<b>Conclusiones finales y trabajos futuros</b>	<b>61</b>
7.1    Conclusiones finales	61
7.2    Trabajos futuros	62
<b>Instalación de MPLAB IDE y MPLAB C Compiler for PIC18</b>	<b>63</b>
<b>Configuración de Proyecto nuevo en MPAB IDE</b>	<b>65</b>
<b>Esquema eléctrico del Hardware</b>	<b>69</b>
<b>Glosario</b>	<b>75</b>
<b>Referencias</b>	<b>77</b>



# Índice de figuras

Figura 1.1 Conceptualización básica del prototipo a diseñar y un segundo nodo.....	2
Figura 2.1 Modelo de Capas OSI – 802.15 .....	8
Figura 2.2 Ejemplos de tipos de topologías de red en 802.15.4.....	11
Figura 3.1 Diagrama de bloques del hardware .....	16
Figura 3.2 Configuración del PIC18F2525 .....	17
Figura 3.3 Conexiones genéricas entre el procesador y el transceptor.....	18
Figura 3.4 Función de inicialización del display .....	20
Figura 3.5 Esquema de funcionamiento de una pantalla táctil resistiva de 4 hilos .....	22
Figura 3.6 Conexiones entre la pantalla táctil y el microprocesador.....	23
Figura 3.7 Conector DB9 hembra con configuración Null-modem .....	24
Figura 3.8 Función de configuración de USART.....	25
Figura 3.9 Fotografía del montaje sobre protoboard .....	26
Figura 3.10 Fotografía del montaje de un Nodo Satélite (Nodo RFD) .....	27
Figura 3.11 Fotografía de la placa definitiva sin las pantallas .....	27
Figura 3.12 Fotografía de la placa definitiva con las pantallas conectadas.....	28
Figura 4.1 Logo de MiWi .....	31
Figura 4.2 Logo de Zigbee .....	32
Figura 4.3 Pila de capas de MiWi (izq.) y Zigbee (drch.) .....	33
Figura 4.4 Ejemplo de topología en estrella de MiWi.....	36
Figura 4.5 Ejemplo de topología punto a punto de MiWi.....	36
Figura 4.6 Tráfico de mensajes bajo petición.....	38
Figura 4.7 Tráfico de mensajes nodo-coordinador.....	39

Figura 4.8 Estructura de canalización de 802.15.4.....	41
Figura 4.9 Estructura del mensaje de red .....	42
Figura 4.10 Estructura del paquete de capa física .....	43
Figura 4.11 Diagrama de funcionamiento de XTEA.....	44
Figura 4.12 Diagrama de bloques de MiWi DE .....	46
Figura 4.13 Ejemplo de configuración de aplicación.....	47
Figura 4.14 Ejemplo de configuración de MiWi P2P .....	49
Figura 5.1 Configuración típica del LM71 5 pines .....	52
Figura 5.2 Diagrama de flujo de la aplicación del coordinador .....	53
Figura 5.3 Pantallas del sistema de menús: vista general.....	54
Figura 5.4 Pantallas del sistema de menús: vista de un nodo.....	54
Figura 5.5 Ejemplo de conversión de dígito a bitmap.....	55
Figura 5.6 Diagrama de flujo de la aplicación del Nodo 2/3.....	56
Figura B.1 Estructura de Archivos en Windows (izq.) y en MPLAB IDE (dcha.) .....	65
Figura B.2 Configuración de las Opciones de Proyecto.....	66
Figura C.1 Esquema eléctrico de las pantallas LCD y táctil .....	70
Figura C.2 Esquema SST de las pantallas LCD y táctil .....	70
Figura C.3 Esquema eléctrico de la CPU .....	71
Figura C.4 Esquema eléctrico del regulador de tensión .....	72
Figura C.5 Esquema SST de la CPU y el regulador.....	72
Figura C.6 Esquema eléctrico del nodo RFD utilizado en las pruebas .....	73

# Índice de tablas

Tabla 1.1 Resumen de las Fases del proyecto. ....	4
Tabla 2.1 Clases de Potencia en Bluetooth.....	9
Tabla 2.2 Velocidad de transmisión de 802.15.3 .....	10
Tabla 2.3 Velocidad de transmisión y modulaciones de 802.15.4 .....	11
Tabla 2.4 Comparación (I) de la familia 802.15.....	13
Tabla 2.5 Comparación (II) de la familia 802.15 .....	14
Tabla 3.1 Funciones de la librería del display .....	20
Tabla 3.2 Funciones de la librería de la pantalla táctil .....	23
Tabla 3.3 Funciones de la librería del módulo USART .....	24
Tabla 4.1 Comparación de MiWi y Zigbee .....	33
Tabla 4.2 Funciones más importantes de MiApp .....	48
Tabla 4.3 Funciones más importantes de MiMAC.....	50
Tabla 5.1 Valores decimales de la temperatura .....	55
Tabla 6.1 Presupuesto del proyecto .....	60





# Capítulo 1:

## Introducción

En el Capítulo inicial de esta memoria, se hace una introducción sobre los tipos de redes inalámbricas y cómo han ido aumentando en importancia y uso en el mercado actual.

A modo de enfoque, este capítulo incluye los objetivos que se persiguen con la realización de este proyecto, así como un historial de las fases del mismo, con una breve descripción de los trabajos acometidos en cada una de ellas.

Para finalizar, se explican los contenidos de cada capítulo para orientar al lector en la lectura de la memoria.

### 1.1 Introducción

En la época actual, la sociedad tiende cada vez más al uso de dispositivos móviles que permitan a los usuarios la posibilidad de continuar comunicados sin la necesidad de permanecer estáticos en un lugar, atado al cable que une el terminal con la red. Para cubrir esta necesidad, la industria ha diseñado múltiples sistemas de comunicaciones inalámbricos, o *wireless*, termino anglosajón muy difundido en el mundo de las comunicaciones.

Los sistemas de comunicación inalámbricos tienen, como principales ventajas frente a las comunicaciones tradicionales cableadas, la movilidad y la flexibilidad del sistema en general. Además, esta clase de sistemas son de menor coste y mayor facilidad y rapidez de instalación debido a que no es necesario ni realizar una gran inversión en infraestructuras ni obras complejas y de larga duración para instalar el cableado. Sin embargo, los sistemas *wireless* también conllevan importantes inconvenientes que se deben tener en cuenta. El espacio radioeléctrico es un bien escaso

y un medio compartido, por lo que los sistemas se ven sometidos a mayores interferencias causadas tanto por otros sistemas, como por factores meteorológicos, topografía del terreno, etc.

Dentro de los sistemas inalámbricos, se debe distinguir el ámbito de aplicación o grado de cobertura del sistema.

- WMAN: acrónimo de *Wireless Metropolitan Area Network*. Redes de cobertura metropolitana.
- WLAN: acrónimo de *Wireless Local Area Network*. Redes de cobertura local.
- WPAN: acrónimo de *Wireless Personal Area Network*. Redes de cobertura personal.
- WBAN acrónimo de *Wireless Body Area Network*. Redes de cobertura corporal.

Este proyecto se plantea para redes WPAN, donde el rango de cobertura oscila entre los 10 y los 75 metros.

## 1.2 Objetivos del proyecto

El objetivo principal del proyecto es realizar un sistema complejo basado en el microcontrolador de referencia utilizado en algunas asignaturas del departamento de Tecnología Electrónica, que incorpore un transceptor que permita comunicaciones inalámbricas. Se persigue crear una placa de desarrollo con: transceptor, display gráfico, pantalla táctil y comunicación serie. Conjuntamente se incluirá un software de demostración para, en un futuro, diseñar unas prácticas que incluyan el uso de los elementos descritos.

También se pretende cumplir con los siguientes objetivos parciales:

- Realizar el software de forma modular, facilitando la inclusión, extracción o modificación de elementos en la placa de desarrollo.
- Diseñar y documentar un sistema de control de temperatura, mediante el uso de varios nodos conectados entre sí.

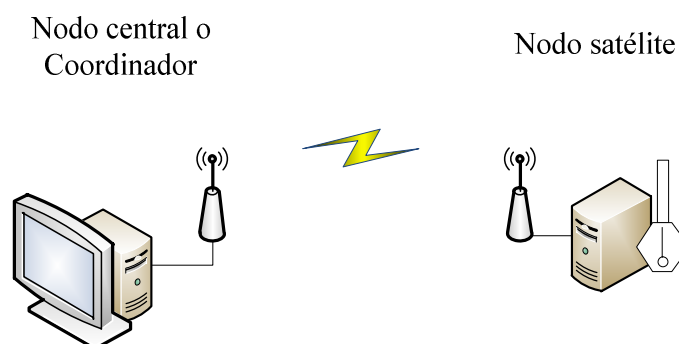


Figura 1.1 Conceptualización básica del prototipo a diseñar y un segundo nodo

## 1.3 Materiales utilizados

A continuación se enumeran los componentes físicos utilizados para la realización del hardware diseñado, tanto para el prototipo final como el usado durante las distintas fases y pruebas realizadas.

- Microcontroladores Microchip PIC18F2525.
- Tarjetas Microchip MRF24J40MA.
- Sensor de temperatura LM71.
- Conector DB9
- Conversor de niveles RS232 ADM3202.
- Regulador de tensión LM317T.
- Pantalla LCD *Displaytech* 64128A.
- Pantalla táctil Nintendo.
- Diversos componentes electrónicos: resistencias, condensadores, diodos, diodos de señal, diodos LED, cristales, cableado, zócalos y pines de conexión.
- Microchip Pickit2 a modo de *debugger* del software.
- Placas de conexión *protoboard*.
- Cable RS232.
- Pila 9V.
- Soldador y estaño
- PC

En cuanto al apartado software, se han utilizado los siguientes elementos:

- Entorno de desarrollo MPLAB IDE v8.66.
- MPLAB C for PIC18 v3.37.01.
- Librerías Microchip v2010-10-19.

Además, se ha hecho uso de los laboratorios docentes del Dto. De Tecnología Electrónica

## 1.4 Fases de desarrollo

En la Tabla 1.1 se resumen las Fases que ha seguido el proyecto, adjuntado una breve descripción y la duración aproximada en la que ha sido desarrollada.

Posteriormente se explican con mayor detalle las tareas llevadas a cabo en cada una de las Fases indicadas.

Fase	Descripción breve	Duración
I	Documentación y conceptualización	2 semana
II	Montaje y prueba de los microprocesadores	1 semana
III	Montaje y prueba de los transceptores y las comunicaciones	4 semanas
IV	Instalación del sensor de temperatura	1 semana
V	Instalación de la pantalla LCD y pantalla táctil	4 semanas
VI	Conexión de las comunicaciones serie	2 semanas
VII	Realización del prototipo definitivo y pruebas en él	8 semana
VIII	Memoria	8 semanas

*Tabla 1.1 Resumen de las Fases del proyecto.*

### ***FASE I: Documentación y conceptualización***

Durante esta fase se ha reunido la documentación relacionada con el hardware que se iría utilizando a lo largo del proyecto, así como se ha esbozado la línea de trabajo a seguir para su desarrollo.

### ***FASE II: Instalación microprocesadores y transceptores***

Montaje del hardware básico de un nodo Coordinador y un nodo satélite, formado por un microcontrolador y su circuito asociado.

### ***FASE III: Comunicaciones inalámbricas***

Acoplar los transceptores MRF24J40 a ambos nodos con las conexiones pertinentes. Adaptar el software de Microchip Inc. que implementa la Pila MiWi al proyecto.

Posteriormente realización de unas pruebas para comprobar que existe comunicación entre los nodos. Dichas pruebas se realizan enviando una secuencia de números de un nodo a otro, mostrando dicho número mediante unos LED conectados para este propósito.

### ***FASE IV: Instalación del sensor de temperatura***

Montaje del sensor LM71 al nodo satélite y comprobar su funcionamiento independiente, comprobando de forma manual que la temperatura obtenida era la real.

### ***FASE V: Instalación de la pantalla LCD y la pantalla táctil***

En esta fase se procede a añadir al nodo central las pantallas que proveerán al proyecto de una interfaz de comunicación entre el usuario y el programa. Después de la instalación física, se implementa el código de inicialización y control de ambas. Durante esta fase también se realizan pruebas de su funcionamiento.

En el nodo central, además de insertar la pantalla, se quitan los LED añadidos durante la Fase III y se realiza el código para mostrar los datos recibidos del nodo satélite por pantalla, comprobando su correcto funcionamiento.

En el nodo satélite se implementa la función para transformar la temperatura de decimal a mapa de bits para su posterior envío al nodo 1, de tal modo que este no tenga que realizar ningún procesado previo para poder ser visualizado correctamente en la pantalla. Este proceso se realiza en el nodo 2 para descargar de esfuerzo computacional al nodo central.

#### ***FASE VI: Conexión de las comunicaciones serie***

Se añade al esquema eléctrico un controlador de comunicaciones serie y un conector para enviar información a un PC mediante RS232. Realizado el montaje, se envían datos desde el procesador al PC y desde el PC al procesador para comprobar que no hay errores en el desarrollo.

#### ***FASE VII: Realización del prototipo definitivo y pruebas en él***

Acabadas todas las conexiones y las pruebas sobre el montaje provisión, se procede a fabricar el prototipo definido con los esquemas eléctricos compuestos por todos los componentes involucrados en el desarrollo del proyecto.

#### ***FASE VIII: Memoria***

Se reúne la documentación del proyecto y se redacta la presente memoria.

## **1.5 Contenido de la memoria**

Esta memoria está compuesta por 7 capítulos y 4 anexos. A continuación se presentan de forma resumida y a modo de orientación, los capítulos y anexos, a excepción del primer capítulo, haciendo una breve descripción de su contenido.

En el 2º capítulo se estudia la familia de estándares 802.15 para redes de área personal y corporal. Al final del mismo, se hace una comparativa de las especificaciones descritas y se dan las razones por las que se ha elegido uno de ellos.

En el capítulo tercero se expone el diseño y fabricación de una placa hardware que sirva como base para la realización futura de prácticas basadas en microcontroladores PIC18F2525. Este capítulo se complementa con el Anexo C, donde se adjuntan los esquemas eléctricos y SST de la placa desarrollada.

Durante el capítulo cuarto se explica el protocolo de comunicaciones inalámbrico elegido, tanto de forma teórica viendo sus características, como de forma práctica, analizando arquitectura y el código de la librería utilizada para su implementación.

En el capítulo quinto, se expone un ejemplo de uso. Durante esta sección se explica la aplicación desarrollada y que hace uso de todos los elementos del hardware diseñado.

El capítulo seis desglosa el coste total del desarrollo y montaje del sistema, incluyendo los beneficios.

En el capítulo siete y último, se recogen las conclusiones tras la finalización de los trabajos, así como posibles trabajos para ampliar la funcionalidad del sistema completo.

Los Anexos A y B indican los pasos a seguir para la instalación del software utilizado, así como para crear y compilar un proyecto que utilice la librería de Microchip.

Complementando a este documento, existe un anexo electrónico con el código implementado para este proyecto.

# Capítulo 2:

## Redes WPAN

En este capítulo, se presentan los estándares integrados dentro de la familia IEEE 802.15, dando los detalles más relevantes para justificar el uso de 802.15.4 en este proyecto.

En el último apartado del capítulo, se agrupa en las Tablas 2.4 y 2.5 las características más importantes de cada una de las especificaciones para facilitar su comparación y se explica la decisión tomada acerca del estándar elegido.

### 2.1 Introducción

La familia de estándares IEEE 802.15 tiene la función de definir un conjunto de especificaciones a nivel físico y de enlace para las conexiones inalámbricas de alcance personal (WPAN) y corporal (WBAN). Cada uno de los estándares del grupo intenta satisfacer las necesidades de segmentos distintos del mercado, cubriendo así comunicaciones que requieren un gran ancho de banda, otras que necesitan transmitir una menor cantidad de datos, redes grandes y pequeñas, etc.

Pese a la distinta orientación de los estándares, de forma común a todos ellos se ha mantenido la idea presentada por el Modelo de Capas de OSI (Figura 2.1) para su diseño y desarrollo.

	Modelo ISO-OSI	Modelo IEEE 802.15
7	Capa de aplicación	Capas superiores
6	Capa de presentación	
5	Capa de sesión	
4	Capa de transporte	
3	Capa de red	
2	Capa de enlace de datos (DLL)	Control de enlace lógico (LLC)
1		Control de acceso a medios (MAC)
	Capa física	Capa física (PHY)

Figura 2.1 Modelo de Capas OSI – 802.15 [1]

El modelo IEEE 802.15 se centra en definir la capa física y la capa de enlace de datos. Tal y como se representa en la figura superior, la capa de enlace de datos se divide en dos subcapas. La subcapa de Control de enlace lógico es común en todos los protocolos del grupo, mientras que la capa MAC varía dependiendo del hardware, y por tanto de la implementación física de la capa.

## 2.2 IEEE 802.15.1

La especificación 802.15.1 [2] [3] es un estándar establecido para regular el protocolo de la industria *Bluetooth*. Por ello, esta especificación es llamada *Bluetooth WPAN* o *Bluetooth*.

Las principales ventajas de este estándar son: una complejidad media-baja, un bajo consumo energético, bajo coste y un alto grado de robustez.

Opera en la banda ISM a 2.4 GHz, por lo que está diseñado para funcionar en entornos con un ruido electromagnético considerable. Para evitar interferencias se utiliza expansión del espectro con saltos en frecuencia, especificando 1600 saltos por segundo entre 79 frecuencias intercaladas cada 1 MHz. El estándar utiliza una modulación GFSK y duplexación en el tiempo, donde cada ranura de tiempo son  $625\mu s$ .

*Bluetooth* tiene dos tipos o modos de funcionamiento: modo síncrono y modo asíncrono, o también llamados modo de conmutación y modo de paquetes,



respectivamente. Las velocidades de transmisión máximas dependen del modo de operación. Para el modo de conmutación, especialmente utilizado para transmisión de voz, se alcanzan los 64 kbps en cada dirección del canal full-duplex. En una comunicación asíncrona la tasa máxima es de, aproximadamente, 1 Mbps.

La potencia de transmisión está dividida en tres clases según se indica en la Tabla 2.1. En función de la potencia, el alcance de las comunicaciones será mayor, siendo necesario el control de potencia en uno de ellos.

Clase de potencia	Potencia (mW)	Alcance (m)
1	100 (Es necesario el control de Potencia)	100
2	2,5	20
3	1	10

Tabla 2.1 Clases de Potencia en Bluetooth

En cuanto a la arquitectura de red, hay que definir el concepto de Piconet. Es una red compuesta por un conjunto de hasta ocho dispositivos *Bluetooth*, permitiendo un único maestro, encargado de definir y notificar la configuración de la conexión (número de saltos, frecuencias a usar, etc.) y siete esclavos. Un mismo terminal puede formar parte de varias piconet, pudiendo así mismo ser maestro en una y esclavo en otra.

Los paquetes de esta especificación tienen una longitud total máxima de 2871 bits, que se dividen en tres partes:

- Código de acceso (68-72 bits): utilizado para sincronización e identificación.
- Cabecera (54 bits): contiene información de control. Es información codificada obtenida a partir de una cabecera de 18 bits.
- Datos (0-2745 bits): información relevante para los niveles/protocolos superiores.

## 2.3 IEEE 802.15.2

El IEEE 802.15.2 [4] no es un estándar de comunicaciones por sí mismo, si no que es una ampliación para posibilitar una mejor coexistencia de *Bluetooth* con otro tipo de redes inalámbricas que funcionan en la banda ISM, específicamente con las WLAN 802.11b. Para ello, en este estándar se definen dos tipos de coexistencia: colaborativa y no colaborativa.

En el caso de coexistencia colaborativa, las colisiones entre los sistemas WPAN y WLAN pueden llegar a ser evitadas si ambos son capaces de compartir información sobre el estado de las transmisiones (velocidad, potencia, etc.). Se detallan tres tipos de coexistencia colaborativa.

En la coexistencia no colaborativa, las colisiones pueden ser evitadas mediante técnicas de censado (escuchar antes de transmitir). Existen cuatro tipos de coexistencia no colaborativa.

Cabe destacar que los métodos colaborativos ofrecen mejores resultados.

## 2.4 IEEE 802.15.3

El estándar 802.15.3 [3] surge de la necesidad de un sistema WPAN capaz de transmitir a altas velocidades de manera eficiente.

Los primeros objetivos a cumplir en este estándar son conseguir bajos costes tanto de implementación como de operación, por ello esta especificación es poco compleja.

Al igual que 802.11b, este estándar, también conocido como *High-Rate WPAN*, opera en la banda ISM. Sin embargo, pueden coexistir debido a que esta especificación ocupa un menor ancho de banda (los canales tienen un BW de 15MHz) y transmite a menos potencia (para una potencia aproximada de 8 dBm, se obtiene un rango de cobertura de entre 30 y 55 metros).

Los rangos de velocidades de transmisión que se alcanzan para este estándar son de 11, 22, 33, 44 y 55 Mbps. Estas tasas dependen del tipo de modulación de la transmisión y de la sensibilidad del receptor según indica la Tabla 2.2.

Modulación	Velocidad (Mbps)	Sensibilidad en el receptor (dBm)
QPSK-TCM	11	-88
DQPSK	22	-81
16-QAM	33	-81
32-QAM	44	-77
64-QAM	55	-74

Tabla 2.2 Velocidad de transmisión de 802.15.3

En relación a la topología, los dispositivos que se comuniquen entre sí forman picoredes, donde se distinguen dispositivos que actúan como PNC, encargados de coordinar todos los dispositivos y aspectos de la picored a la cual pertenecen (solo puede existir uno por picored), o como DEV, que son los que simplemente transmiten información.

La trama básica de este estándar apenas sobrecarga la transmisión, ya que solamente introduce un preámbulo, usado para información de gestión (duración de la trama, canales asignados a un determinado DEV, etc.), y un control de acceso periódico basado en CSMA/CA, lo que permite una mayor eficiencia temporal y baja latencia.

Posteriormente al 802.15.3, se han ido desarrollando modificaciones a éste y han aparecido nuevas especificaciones: 802.15.3a, 802.15.3b y 802.15.3c. Estas actualizaciones han tenido mayor éxito en el mercado, y algunas de sus características son:

- 802.15.3a: se cambia la banda de transmisión a una banda entre 3.1 y 10.6 GHz. Se consiguen mayores velocidades para un alcance no superior a la decena de metros.
- 802.15.3b: trabaja para mejorar la capa MAC del protocolo original.

- 802.15.3c: trabajando en la banda de 57 a 64 GHz, se conseguirían velocidades en torno a los 2 Gbps.

## 2.5 IEEE 802.15.4

El estándar 802.15.4 [3] [5] es desarrollado para cubrir las comunicaciones que no requieren de una alta velocidad de transmisión. Esta especificación también es conocida como *Low-Rate WPAN*.

Fue desarrollado con la idea de ser un estándar de bajo coste de implementación y operación, baja potencia de emisión y sistemas poco complejos.

El estándar 802.15.4 admite variedad de arquitecturas para su conexión en red. Esta topología, por ejemplo en estrella o punto a punto, es una elección durante el diseño de las funciones de la red y la aplicación que en un futuro vaya a tener.

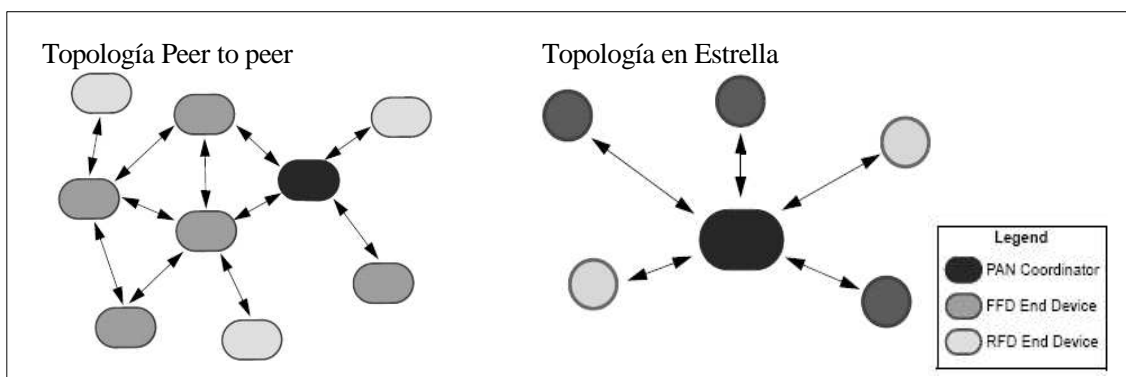


Figura 2.2 Ejemplos de tipos de topologías de red en 802.15.4

En la Capa Física se han definido tres bandas de operación con un distinto número de canales en cada una de ellas. Para la banda situada en 868 MHz 1 único canal; para la banda en 915 MHz 10 canales; y para la banda ISM 16 canales (situados cada 5 MHz en las frecuencias entre 2.4 GHz – 2.4835 GHz).

Los parámetros de velocidad y modulación según la banda de operación quedan recogidos en la Tabla 2.3.

Banda	Velocidad (kbps)	Modulación
868 MHz	20	BPSK
915 MHz	40	BPSK
2,4 GHz	250	O-QPSK

Tabla 2.3 Velocidad de transmisión y modulaciones de 802.15.4

La capa MAC del estándar 802.15.4 se encarga de la asociación y disociación, del reconocimiento y validación de las tramas recibidas, garantizar el uso correcto de las ranuras de tiempo de transmisión y del manejo de guías.

Ha sido diseñada salvaguardando el objetivo de lograr la mayor simplicidad y flexibilidad del estándar y únicamente define 26 primitivas para su administración (número sensiblemente inferior a las, que por ejemplo, 131 primitivas que maneja *Bluetooth*).

El formato de mensaje general mantiene una forma sencilla, conservando la flexibilidad adecuada para adaptarse a multitud de aplicaciones y usos a los que el estándar pueda estar destinado.

Otras características generales de la especificación IEEE 802.15.4 son:

- Alcance hasta 75 metros a potencia típica.
- Detección de los niveles de energía recibidos.
- Direccionamiento de 16 (corta) o 64 bits (larga, IEEE).
- Indicadores de calidad en el enlace.
- Acceso al canal por CSMA/CA ranurado o no ranurado.
- Latencia inferior a los 15 ms.
- Puede operar en un modo opcional de Súper-estructuras (supertramas).
- Uso de tramas ACK.
- Establecimiento de la red automáticamente por el coordinador.
- Permite elegir entre el uso de tramas *beacon*<sup>1</sup> o no.
- Provee de seguridad.

Con el paso de los años, el IEEE ha trabajado en revisiones y ramificaciones de la especificación, poniendo al final del nombre una letra. En este documento se hace referencia siempre a datos de la versión original.

## 2.6 IEEE 802.15.5

Este grupo [3] no es un estándar en sí, al igual que 802.15.2. Surge en 2003 para posibilitar al resto de especificaciones trabajar con una topografía de malla.

Se definen la malla total, en la que cualquier nodo puede estar conectado con cualquier otro dispositivo; y la malla parcial, en el que solo algunos nodos pueden conectarse con cualquiera del resto.

Las características de esta topografía son:

- Posibilidad de llevar los mensajes por distintos caminos.
- No pueden existir interrupciones de las comunicaciones.
- No requiere nodo central, por lo que reduce el mantenimiento.

---

<sup>1</sup> La trama beacon es un tipo especial de trama que se envía periódicamente o bajo petición con información sobre la red.

- Si un nodo desaparece, el resto se ve menos afecto o no afectado que con otras topologías.

## 2.7 Otros estándares 802.15: 802.15.6 y 802.15.7

El estándar 802.15.6 [3], aún siendo de la misma familia, está considerado una especificación de red corporal, WBAN, con un alcance en torno al metro. Por tanto no ha sido considerado y ha sido obviado en el desarrollo de esta memoria.

El estándar 802.15.7 [6] define las comunicaciones mediante luz visible. Esta tecnología proporciona grandes anchos de banda para alcances cortos (hasta 100 Mbps en distancias inferiores a los 3 metros). Además de la velocidad de transmisión, su gran aliciente es la reducción drástica en el consumo de energía.

En la actualidad se trabaja en la reducción de costes, aumentar el alcance e interconexionar con otros sistemas inalámbricos tradicionales.

## 2.8 Comparativa y conclusiones

Cada uno de los estándares descritos anteriormente, tiene unas características muy diferentes e intentan cubrir todas las necesidades de la industria en cuando a redes de área personal se refiere.

En la Tabla 2.4 se resumen las características físicas más relevantes a la hora de elegir el estándar apropiado a los trabajos que aquí se describen.

Estándar	Velocidad	Potencia (mW)	Alcance máximo (m)	Banda de frecuencia	QoS
802.15.1	1 Mbps	100, 2-5, 1	100	2.4 GHz	Media
802.15.3	11, 22, 33 o 55 Mbps	6	50	2.4 GHz	Alta
802.15.3a	400 Mbps	1	10	3.1-10.6 GHz	Muy alta
802.15.4	20, 40, 250 Kbps	1	75	868.915 MHz y 2.4 GHz	Baja

Tabla 2.4 Comparación (I) de la familia 802.15

En la Tabla 2.5 se resumen las características de implementación y mantenimiento del sistema.

Estándar	Complejidad	Coste	Eficiencia	Consumo
805.15.1	Media-Alta	Bajo	Media	Alto
805.15.3	Alta	Medio	Alta	Medio-Alto
805.15.3a	Alta	Medio	Alta	Medio-Alto
805.15.4	Baja	Bajo	Muy alta	Muy bajo

*Tabla 2.5 Comparación (II) de la familia 802.15*

Debido a las caracterizas docentes del proyecto, el sistema a desarrollar tendrá que tener el menor coste y consumo posible. Además, el manejo y uso del estándar de comunicaciones deberá ser sencillo y flexible para que el desarrollador pueda centrarse en la implementación de la aplicación en lugar de involucrase en exceso con los problemas del protocolo de comunicación inalámbrica. Por otro lado, la aplicación no necesitará de un gran ancho de banda para las comunicaciones con otros nodos, teniendo un número limitado de éstos conectados.

Es por estas razones que el presente proyecto define el IEEE 802.15.4 como el estándar más acorde con los objetivos definidos en el capítulo anterior.

## **Capítulo 3:**

# **Diseño e implementación del prototipo**

En este capítulo se expone detalladamente la composición del hardware desarrollado, así como su diseño, implementación y pruebas de funcionamiento. En primer lugar se verá el conjunto de forma global mediante el diagrama de bloques del sistema, con una breve explicación de la funcionalidad esperada de cada elemento. A continuación, se analizará cada componente por separado, profundizando en su funcionamiento detallando el software desarrollado, incluyendo las funciones de la librería que operada cada uno.

Durante el proceso de diseño del hardware se ha utilizado el montaje de los circuitos en placas Protoboard para facilitar los sucesivos cambios y mejoras que se realizaban para corregir u optimizar el comportamiento de los dispositivos. Una vez realizadas todas las pruebas y comprobar que los esquemas eran correctos, se ha procedido al diseño del prototipo definitivo sobre las placas de circuito impreso con la colaboración de la Oficina técnica de la UC3M.

### **3.1 Diagrama de bloques**

En la siguiente figura se representa el diagrama de bloques utilizado como base para la implementación del hardware.

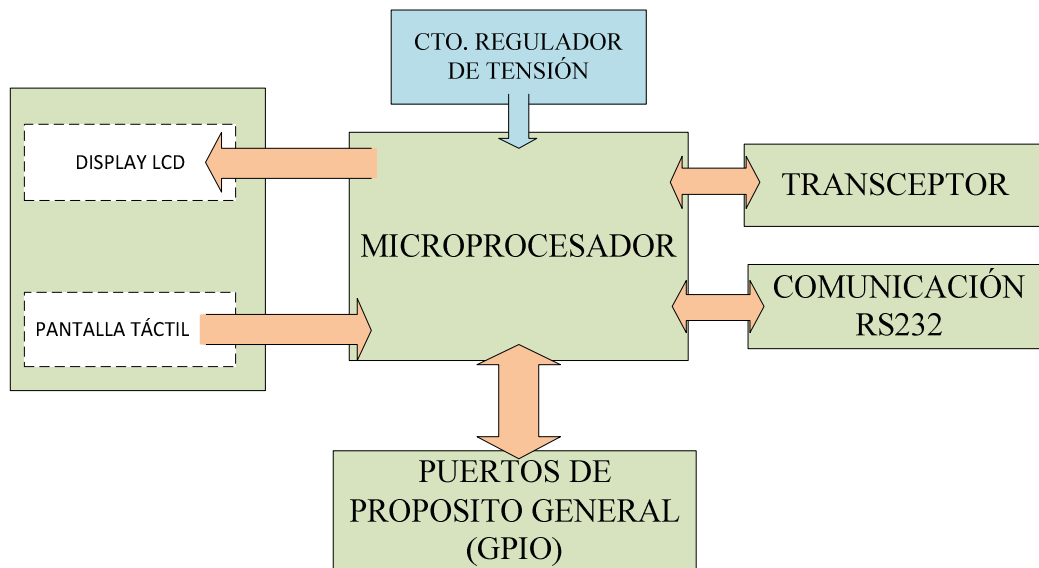


Figura 3.1 Diagrama de bloques del hardware

El microprocesador utilizado será un PIC18F2525 y será el núcleo del prototipo. El micro se encargará de ejecutar el software de la aplicación, configurar, controlar y utilizar el resto de los dispositivos que componen la totalidad del prototipo.

Para las comunicaciones inalámbricas con otros nodos, se usará la tarjeta MRF24J40MA de Microchip, que incluye el controlador de las comunicaciones, una antena PCB y el circuito asociado.

De forma complementaria, las comunicaciones serie RS232 proporcionan una forma sencilla de intercambio de información con un PC, mediante el uso de un programa de comunicación serie (*HyperTerminal* o *Serial Port Monitor*, por ejemplo).

Para implementar una interfaz con el usuario sin necesidad de PC, la placa de desarrollo incorpora un *display* gráfico para mostrar información, y una pantalla táctil con la cual el usuario podrá realizar peticiones al microprocesador a través de un sistema de menús que implementará la aplicación.

Además de estos dispositivos, la placa deberá ser lo suficientemente flexible como para poder incorporar nuevos dispositivos hardware para ampliar su funcionalidad. Para ello, se extraen y se dejan accesibles al usuario los pines del procesador que no hayan sido utilizados, así como otros puertos claves, como el SPI.

De forma independiente, el modelo final contará con un circuito regulador de tensión para permitir alimentación externa y minimizar el efecto perjudicial en los dispositivos de picos de tensión.

## 3.2 Microcontrolador PIC18F2525

El PIC18F2525 de Microchip [7] es un microcontrolador de bajo coste con las suficientes características para dar una correcta funcionalidad al proyecto. El micro será el centro del prototipo y desde él se controlará todos los demás elementos.



Sobre el hardware, el PIC requiere inicialmente de un pequeño circuito de alimentación, oscilador externo, reset y toma a tierra. Dicha alimentación externa se realiza a través de un circuito regulador de tensión, que además de proporcionar el voltaje necesario para que funcionen todos los componentes, protege el sistema de picos de tensión que puedan dañar los instrumentos. La placa incluye un oscilador externo permitiendo elegir en la configuración del procesador si se desea utilizar este o usar el oscilador interno. El circuito de reset, una vez habilitado el reset por hardware en la configuración, servirá para reiniciar la ejecución del código programado en el PIC. Además, se incluye una tira de pines para conectar el procesador al PC mediante el PicKit 2. Con este instrumento se podrá depurar el programa mientras se ejecuta el código sobre el propio microprocesador. Además permitirá programarlo para operar con la placa de desarrollo únicamente conectado una alimentación externa.

Todas estas conexiones pueden consultarse en el esquema eléctrico de la placa de desarrollo del Anexo C.

En cuanto al software, el PICF2525 requiere una configuración inicial, siendo optativa en caso de querer usar la configuración por defecto. Los aspectos más importantes se muestran en la figura siguiente.

```

/***** CONFIG *****/
#pragma romdata CONFIG1H = 0x300001
    const rom unsigned char config1H = 0b00000110;    // Oscilador en modo HSPLL

#pragma romdata CONFIG2L = 0x300002
    const rom unsigned char config2L = 0b00011111;    // Brown-out Reset habilitado por hardware
                                                    // PWRTEN deshabilitado

#pragma romdata CONFIG2H = 0x300003
    const rom unsigned char config2H = 0b00010110;    // HW WD deshabilitado, 1:32 prescaler

#pragma romdata CONFIG3H = 0x300005
    const rom unsigned char config3H = 0b10000000;    // Puerto 1 para RESET

#pragma romdata CONFIG4L = 0x300006
    const rom unsigned char config4L = 0b01000001;    // DEBUG habilitado(0)/disabled(1)
                                                    // XINST habilitado - modo extendido
                                                    // LVP deshabilitado
                                                    // STVREN deshabilitado

```

Figura 3.2 Configuración del PIC18F2525

En primer lugar, se configura para usar el oscilador externo en modo HS, al que además se activa el modo PLL según se recomienda en la librería usada para implementar el protocolo de comunicaciones.

También hay que habilitar tanto el reset por hardware y como el pin 1 para dicho cometido.

Es importante habilitar el modo *debugg* para poder ejecutar y depurar el código directamente sobre el procesador. Otra de las configuraciones requeridas por las comunicaciones es usar el modo extendido del procesador, poniendo el bit XINST a 1 tal y como se indica en la hoja de características del procesador.

Con la configuración finalizada, se han incorporado el resto de elementos. En los apartados siguientes se explica detalladamente cada elemento, tratando en cada caso los pines del PIC18F2525 de los que hacen uso.

### 3.3 Transceptor MRF24J40MA

MRF24J40MA [8] es la nomenclatura usada por el fabricante para indicar la tarjeta compuesta por el controlador MRF24J40 [9], su circuito asociado y una antena PCB. A dichas tarjetas, y con el fin de poder conectarlas a la placa, se le ha soldado un pin en cada uno de los puertos

El transceptor es el dispositivo que otorga a la placa de desarrollo de la capacidad de comunicación inalámbrica y cuenta con la ventaja de haber sido diseñado especialmente para protocolos basados en IEEE 802.15.4, aunque únicamente es capaz de operar en la banda ISM.

La conexión entre el transceptor y el microprocesador se realiza mediante el módulo MSSPI operando en SPI del PIC y la comunicación SPI de la tarjeta. De este modo, se conectarán los pines 14 (SCK), 15 (SDI) y 16 (SDO) a sus homólogos en la tarjeta MRF24J40MA.

Tal y como se muestra en la Figura 3.3, el transceptor cuenta también con 3 pines de control, que permitirán habilitar o deshabilitar el uso del mismo (pin CS), reiniciarlo (pin RESET) y cambiar de estado inactivo a activo (pin WAKE). Estos tres pines se conectan con cualquier puerto genérico del procesador, si bien en el software adjunto a esta memoria se han utilizado RC0, RC1 y RC2 para CS, WAKE y RESET, respectivamente.

Finalmente, el transceptor incorpora un pin de interrupción para avisar al PIC de la existencia de nuevos mensajes. Esta conexión puede realizarse a cualquier pin de interrupción del microprocesador. En este caso se ha utilizado el pin INT1, respetando la configuración recomendada en las librerías y documentos de Microchip.

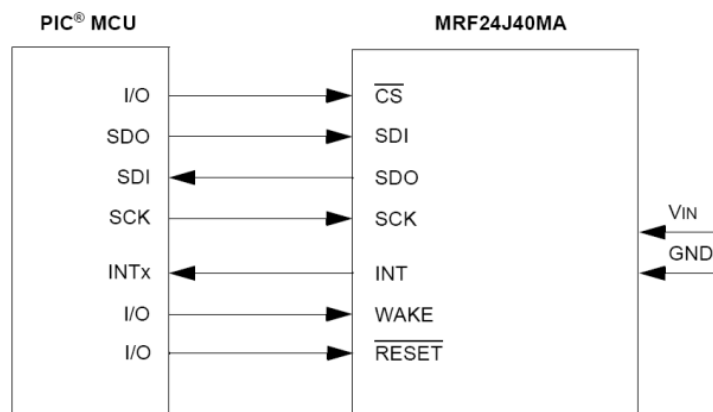


Figura 3.3 Conexiones genéricas entre el procesador y el transceptor

Debido a la utilización de las librerías de Microchip para las comunicaciones, no se ha programado las funciones para utilizar la tarjeta. El software utilizado se expone en el apartado 4.5.

### 3.4 Pantalla LCD

La pantalla utilizada es un display gráfico monocromo monocromo de marca *Displaytech* [10] con una resolución de 128x64 píxeles, que incorpora un controlador KS0713 [11].

Este modelo de pantalla puede ser conectado al microprocesador en serie, utilizando la comunicación SPI, o en paralelo, usando los puertos de propósito general (GPIO). Con una configuración en paralelo, la aplicación es capaz tanto de enviar datos a la pantalla como leer los registros de información de la misma. A cambio de esta capacidad, el procesador necesita utilizar, ya sea en exclusiva o de forma compartida, 9 pines más que si la configuración fuera en serie (en su *datasheet*, los puertos DB0 a DB5, E, R/W y MI, que en serie quedan libres o a alimentación). En este último modo, se pueden enviar datos, pero no recibirlos, limitando su funcionalidad. Debido al reducido número de pines del PIC18F2525 y para no complicar en exceso la interconexión de dispositivos compartiendo pines, el prototipo se ha realizado para usar el *display* con una configuración serie.

Con dicha configuración, se requiere el uso de los pines 14 (SCK) y 16 (SDO) del procesador para las comunicaciones SPI de sentido único (desde el procesador a la pantalla). Además, la pantalla utilizará otros tres puertos de propósito general para las señales de *reset*, de *Chip Select* y de *switch* para diferenciar entre datos de configuración y datos a representar. Para este diseño, se han utilizado los puertos 23 (RB2), 24 (RB3) y 25 (RB4) para el *Chip Select*, *reset* y *switch*, respectivamente.

Aunque la pantalla y el transceptor hacen uso del pin 16 del PIC18F2525, no entran en conflicto debido a que cuando se utilice uno, el otro estará deshabilitado mediante su puerto CS.

En base los diagramas de flujo y detalle de los comandos aceptados por el controlador expuestos en su *datasheet* (Tabla 18 y Figura 27, entre otras) se han definido las variables y las funciones de la librería implementada para el uso del DISPLAY correctamente.

En la siguiente figura se presenta el contenido de la función de inicialización de la pantalla. Para más información sobre las variables de configuración o las funciones presentadas en la tabla 3.1, se puede consultar el código adjunto a la memoria, concretamente en los archivos Lcd.h y Lcd.c, presentes dentro de la carpeta Pantallas.

```

void LcdResetInit() {

    LCD_RESET = 0;
    _delay_us( LCD_DELAY );
    LCD_RESET = 1;
    _delay_us( LCD_DELAY );

    LcdEnviarComando( LCD_CMD_ON, 1 );

    LcdEnviarComando( LCD_CMD_ADC, 0 );
    LcdEnviarComando( LCD_CMD_SHL, 0 );
    LcdEnviarComando( LCD_CMD_BIAS, 1 );

    LcdEnviarComando( LCD_CMD_POWERC, 7 );
    _delay_us( LCD_DELAY );

    LcdEnviarComando( LCD_CMD_RES, 5 );
    LcdEnviarComando( LCD_CMD_SRV, 1 );
    LcdEnviarComando( 0x00, 5 );

    LcdEnviarComando( LCD_CMD_SIR, 1 );
    LcdEnviarComando( 0x00, 3 );
    LcdEnviarComando( LCD_CMD_REVERSE, 0 );
    LcdEnviarComando( LCD_CMD_IDL, 0 );
    LcdEnviarComando( LCD_CMD_EON, 0 );
}

```

Figura 3.4 Función de inicialización del display

Las funciones de la figura anterior forma parte de un conjunto de once funciones definidas en total.

Nombre	Descripción	Parámetros de entrada	Tipo
LcdBorrar	Borra una o todas las líneas de la pantalla	short fila - fila a borrar (valor 10 para borrar todas)	void
LcdDibujarListado	Escribe la lista de nodos conectados	short cuentaNodos - número de nodos conectados u08 *T_Nodo_A - T° en el Nodo A u08 *T_Nodo_B - T° en el Nodo B u08 *T_Nodo_C - T° en el Nodo C u08 *T_Nodo_D - T° en el Nodo D	void
LcdDibujarSubMenu	Escribe la ventana de un nodo particular	rom u08 *nodo - representación bit-map de la letra del nodo u08 *T_Nodo - temperatura del nodo	void

Tabla 3.1 Funciones de la librería del display

Nombre	Descripción	Parámetros de entrada	Tipo
LcdEnviarComando	Envía un comando de configuración al LCD	u08 comando - byte por defecto de cada comando u08 flag - bits de configuración de cada comando	void
LcdEscribirByte	Envía un byte para ser mostrado por pantalla	u08 datos - byte a representar	void
LcdEscribirGrafico	Dibuja un gráfico (conjunto de bytes) guardado en RAM de una línea en el LCD	u08 *graf - puntero a un conjunto de datos escritos en RAM	void
LcdEscribirGraficoRom	Dibuja un gráfico (conjunto de bytes) guardado en ROM de una línea en el LCD	rom u08 *graf - puntero a un conjunto de datos escritos en ROM	void
LcdLinea	Dibuja una línea horizontal en la pantalla	u08 ini_x - coord. X de inicio u08 ini_y - coord. Y de inicio u08 fin - coord. X de fin	void
LcdPunto	Enciende un pixel de la pantalla	u08 x - coord. X u08 y - coord. Y	void
LCDResetInit	Resetea e inicia la configuración del LCD	Ninguno	void
LcdSetPosicion	Modifica la posición del cursor de escritura	u08 fila - fila de la nueva posición u08 columna - columna de la nueva posición	void

Nota: u08 = unsigned char

Tabla 3.1 Funciones de la librería del display (Continuación)

Durante el desarrollo y las pruebas del hardware y software, se ha utilizado una pantalla *DisplayTech* modelo 64128A existente en los laboratorios del departamento. Sin embargo, esta pantalla actualmente se encuentra descatalogada por lo que no se podrían adquirir nuevas unidades. Para solventar este problema, se recomienda usar una pantalla *DisplayTech* modelo 64128M [12] [13], dispositivo cuyo controlador es completamente compatible con las librerías del aquí presentadas y usadas durante las pruebas. Únicamente requiere una pequeña reconfiguración hardware (cambio de 30 a 28 pines).

## 3.5 Pantalla táctil

Como interfaz entre el usuario y la aplicación, el prototipo incorpora una pantalla táctil resistiva de cuatro hilos [14]<sup>2</sup>. Este tipo de pantallas táctiles es uno de los tipos más extendidos, económicos y de fácil manejo.

En este tipo de dispositivo, cada hilo está conectado a un lado, tal y como se refleja en la Figura 3.5. Cuando la pantalla está correctamente alimentada por el procesador, los hilos del lado izquierdo y lado inferior varían su voltaje en función del lugar en el que la pantalla esté siendo pulsada.

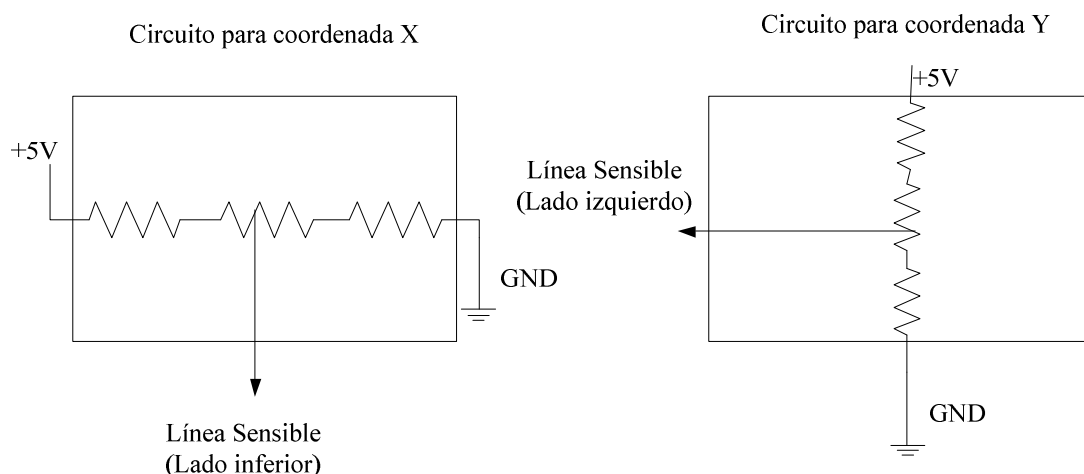


Figura 3.5 Esquema de funcionamiento de una pantalla táctil resistiva de 4 hilos

Pese a que el voltaje recomendado de alimentación de la pantalla táctil es de 5V, en la placa de desarrollo será a 3.3V para hacerlo compatible con otros elementos que no soportan voltajes superiores a 3.6V.

La lectura de los voltajes de salida se realiza mediante el módulo de conversión A/D del PICF2525. Este módulo convierte una señal analógica de entrada a un valor digital que se almacenará en dos registros dispuestos para este uso. El procedimiento de lectura requiere de un tiempo de adquisición, tras el cual se procede a realizar la conversión. Una vez realizada la conversión, hay que esperar de nuevo para que el condensador del módulo se descargue y se pueda realizar el proceso otra vez.

La pantalla, por lo tanto, se conecta a 4 puertos del Conversor A/D del procesador. Cada par de hilos va asociado a una coordenada, de tal modo que, configurando los hilos como entrada/salida, con voltaje/sin voltaje y a tierra, podremos conseguir la conversión a digital del voltaje medido tras presionar un punto concreto de la pantalla.

Para el manejo de la pantalla táctil, primeramente en el procesador hay que definir qué puertos vamos a utilizar. El PIC18F2525 tiene el Conversor A/D en los Puertos A. La forma más intuitiva de conectarlo es usando 4 puertos RA, aunque únicamente 2 de ellos tendrán una función de conversión propiamente dicha (los pines conectados al lado inferior e izquierdo de la pantalla). Teniendo esto en cuenta y

<sup>2</sup> Este documento muestra el funcionamiento de una pantalla táctil resistiva de 4 hilos genérica, por lo que hay que asegurarse de definir correctamente la relación entre cada hilo y cada lado de la pantalla.

pudiendo haber usado RA0, RA1 y otros dos GPIO libres, se han utilizado y conectado los pines tal y como se indica en el extracto de código mostrado por la Figura 3.6.

```
//Configuración TouchScreen
#define TS_RIGHT_1          LATAbits.LATA3      // Hilo 1 del TS - Derecha
#define TS_RIGHT_1_TRIS    TRISAbits.TRISA3
#define TS_BOTTOM_2        LATAbits.LATA2      // Hilo 2 del TS - Abajo
#define TS_BOTTOM_2_TRIS   TRISAbits.TRISA2
#define TS_LEFT_3           LATAbits.LATA1      // Hilo 3 del TS - Izquierda
#define TS_LEFT_3_TRIS     TRISAbits.TRISA1
#define TS_TOP_4            LATAbits.LATA0      // Hilo 4 del TS - Arriba
#define TS_TOP_4_TRIS      TRISAbits.TRISA0
```

Figura 3.6 Conexiones entre la pantalla táctil y el microprocesador

Tras estas definiciones, hay que configurar los registros que controlan las opciones del Conversor A/D según se indica en el *datasheet* del procesador. En este caso, habrá que habilitar la conversión para los canales 1 y 2. Realizados estos pasos, se ha implementado la librería de funciones que permiten el uso de la pantalla táctil. Estas funciones se encuentran en los archivos TouchScreen.h y TouchScreen.c, de la carpeta Pantallas. La configuración inicial del conversor se realiza en HardwareProfile.c.

Función	Descripción	Parámetros de entrada	Tipo
TsCoorX	Captura el voltaje de la posición X en la pantalla táctil	Ninguno	unsigned char - devuelve el valor de la conversión en 1 byte
TsCoorY	Captura el voltaje de la posición Y en la pantalla táctil	Ninguno	unsigned char - devuelve el valor de la conversión en 1 byte

Tabla 3.2 Funciones de la librería de la pantalla táctil

## 3.6 Comunicación RS232: ADM3202 y conector DB9

El interfaz RS232 se implementa con el módulo USART del microprocesador, usando los pines 17 (RC6) y 18 (RC7) para transmisión y recepción de datos, respectivamente. Al PIC18F2525 se le conecta un adaptador de niveles ADM3202 [15] y un conector DB9 hembra. La placa de desarrollo implementa la conexión null-modem sin control de flujo. En el esquema eléctrico de la placa del Anexo C se puede consultar las conexiones entre estos elementos y los condensadores externos que requiere el ADM3202.

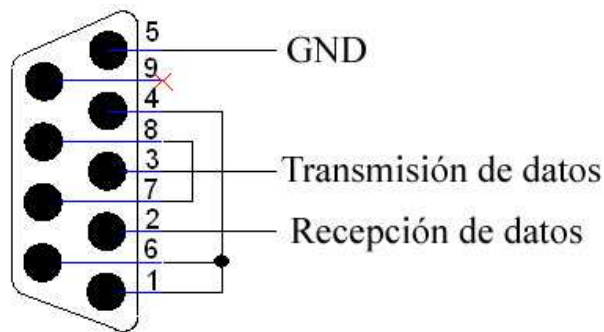


Figura 3.7 Conector DB9 hembra con configuración Null-modem

Realizadas las conexiones físicas, hay que configurar el módulo USART de forma apropiada, seleccionando las características de la comunicación serie: velocidad de transmisión en baudios por segundo, bit de paridad, bit de parada y uso de 8 o 9 bits de información. De forma complementaria se puede habilitar el uso de interrupciones de recepción, de envío o de ambas.

Para lograr la funcionalidad completa del USART, la librería incluida en el software implementa cinco funciones: de inicialización, de escritura de carácter (indistintamente RAM y ROM, de escritura de cadena desde RAM y ROM y de lectura de carácter. Dicha librería puede consultarse en el Anexo electrónico de esta memoria, concretamente en los archivos USART.h y USART.c, incluidos en la carpeta Comun.

Función	Descripción	Parámetros de entrada	Tipo
USARTecribirByte	Envía un byte por USART	char byte - byte a enviar por el USART	void
USARTecribirCadena	Envía una cadena de bytes almacenados en RAM por USART	char *cadena - cadena de caracteres almacenada en RAM a enviar por el USART	void
USARTecribirCadenaR	Envía una cadena de bytes almacenados en ROM por USART	const rom char *cadena - cadena de caracteres almacenada en ROM a enviar por el USART	void
USARTInit	Inicializa los valores del USART a 9600 8N1 e interrupción de recepción activada	Ninguno	void
USARTLeerByte	Recoge un byte recibido por el USART	Ninguno	char - devuelve el carácter recibido por el USART

Tabla 3.3 Funciones de la librería del módulo USART



En la siguiente figura se muestra la función USARTInit, que configura el modulo USART de forma predeterminada a los valores de comunicación serie más habituales: 9600 baudios, 8 bits, sin bit de paridad, 1 bit de parada (9600 8N1), y únicamente activada la interrupción por recepción.

```
void USARTInit() {

    //Configuración puertos para USART
    TRISCbits.TRISC6 = 1;
    TRISCbits.TRISC7 = 1;

    BAUDCONbits.BRG16=0; // Baudrate 8 bits
    TXSTAbits.BRGH = 1; //Alta velocidad
    SPBRG = 204; // = (XTAL/(16*baudrate))-1
    // Baudrate = 9600
    // Oscilador en modo HSPLL = Fosc x 4 = 8MHz x 4 = 32 MHz

    RCSTAbits.SPEN = 1;
    TXSTAbits.SYNC = 0; //Modo asíncrono
    TXSTAbits.TXEN = 1; //Habilitado recepción y envío
    RCSTAbits.CREN = 1;
    TXSTAbits.TX9 = 0;
    RCSTAbits.RX9 = 0;

    //INTERRUPCION
    PIE1bits.RCIE = 1;
    INTCONbits.PEIE = 1;
    INTCONbits.GIE = 1;

}
```

*Figura 3.8 Función de configuración de USART*

Hay que mencionar que el compilador C18 incluye una librería para usar el periférico USART, evitando tener que programarla por el usuario. Pese a ello, se ha decidido incluir las funciones propias y no la librería del compilador por los siguientes motivos:

- Mayor independencia de la versión del compilador.
- El código de las funciones es más accesible.

## 3.7 Regulador de tensión LM317

Para evitar picos y variaciones de tensión indeseadas, el prototipo incluye un circuito regulador de tensión con un LM317 [16]. Este regulador necesita una alimentación de 3 voltios por encima del voltaje deseado a la salida, por lo que, para una alimentación deseada de 3.3 V, hay que alimentar la placa de desarrollo con, al menos, 6.3 V, siendo este último el valor recomendado. En el Anexo C puede consultarse el esquema eléctrico del circuito regulador.

### 3.8 Prototipos

Como ya se comentó en los párrafos iniciales del capítulo, antes de realizar el montaje definitivo sobre las placas de circuito impreso, se han utilizado placas *protoboard* para conectar de forma temporal todos los elementos y permitir realizar todas las pruebas necesarias para comprobar su correcto funcionamiento, así como facilitar los cambios necesarios en las conexiones, ya sea por correcciones o por cambios de pines por meros motivos de comodidad a la hora de diseñar los recorridos de las pistas de conexión en la placa definida que montará el hardware descrito.

La siguiente figura muestra una fotografía del montaje temporal explicado, indicando la posición de cada elemento.

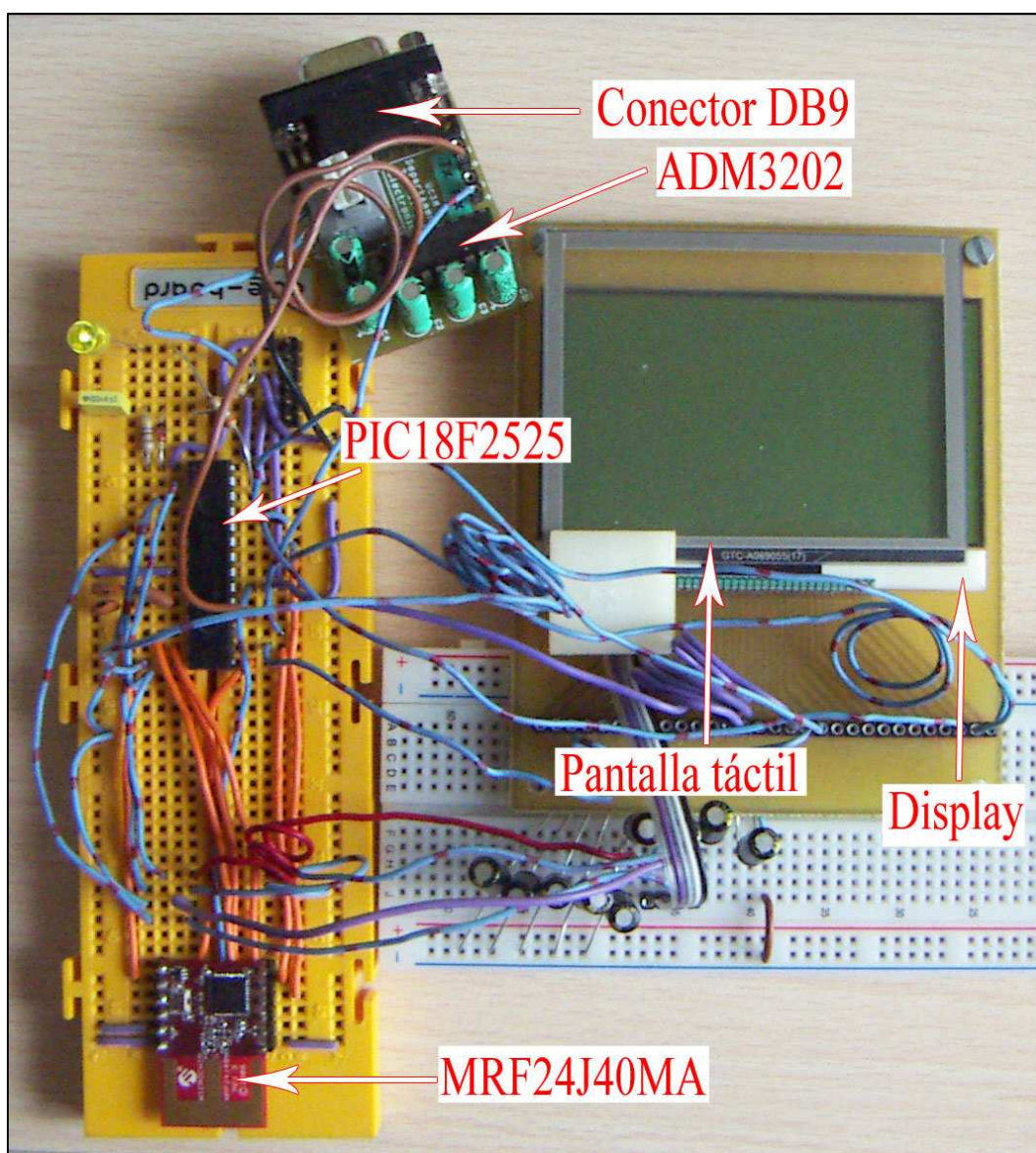


Figura 3.9 Fotografía del montaje sobre protoboard

De forma independiente a la placa de desarrollo, se ha construido un sistema únicamente con un PIC18F2525, un transceptor y un sensor de temperatura LM71 para realizar un ejemplo de aplicación o práctica de la placa objeto de este proyecto. En el capítulo cinco se puede leer más sobre esta aplicación.

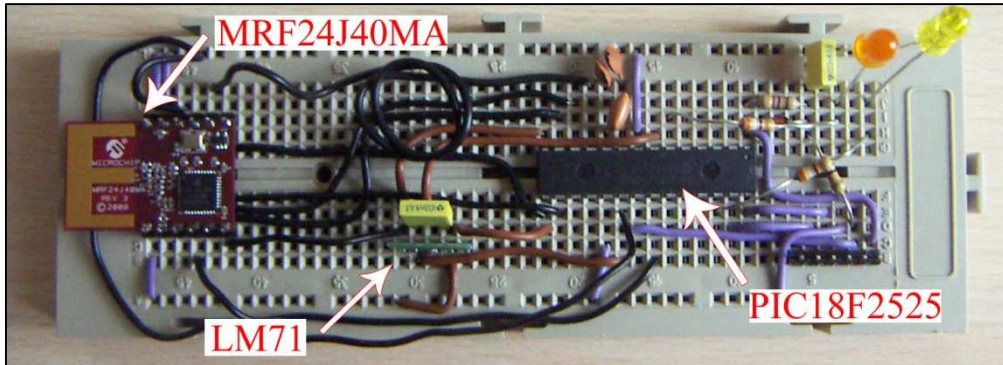


Figura 3.10 Fotografía del montaje de un Nodo Satélite (Nodo RFD)

Después de realizar las pruebas sobre el modelo de pruebas y comprobar que todos los elementos funcionan perfectamente, se ha procedido con la ayuda de la Oficina Técnica a la fabricación de un prototipo de la placa definitiva. El montaje se ha realizado siguiendo los esquemas eléctricos presentados en el Anexo C, donde se pueden consultar los esquemas eléctricos dibujados en OrCAD de las dos placas definitivas (una que monta las pantallas y otra con el resto de componentes y el circuito de alimentación) así como los esquemas de distribución de los elementos en las placas.

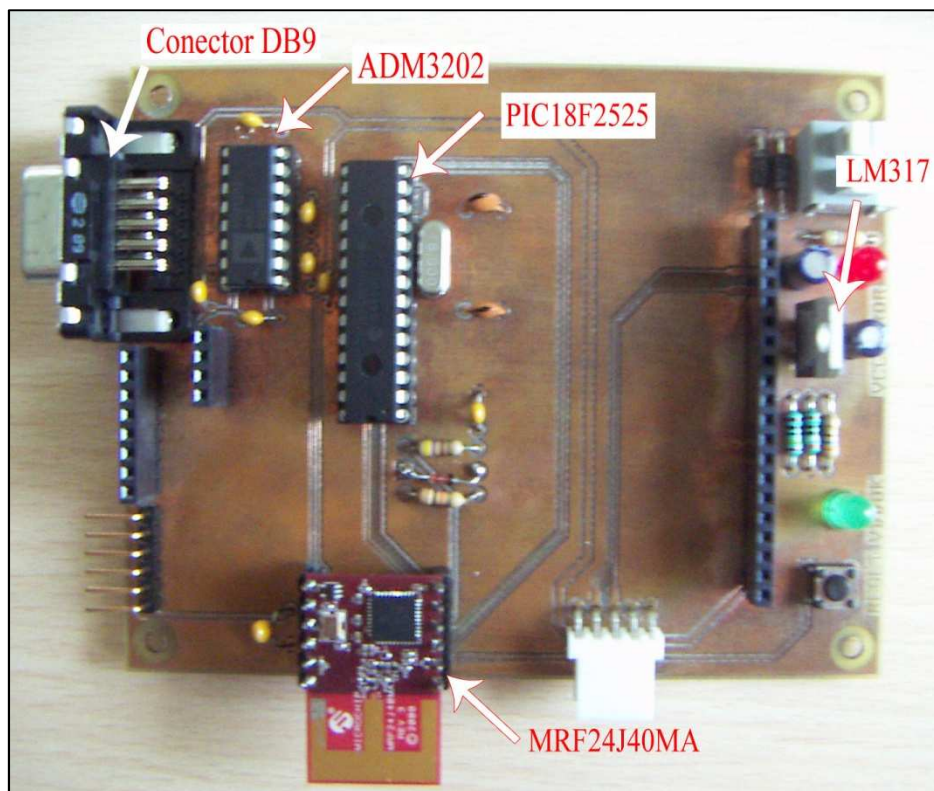


Figura 3.11 Fotografía de la placa definitiva sin las pantallas



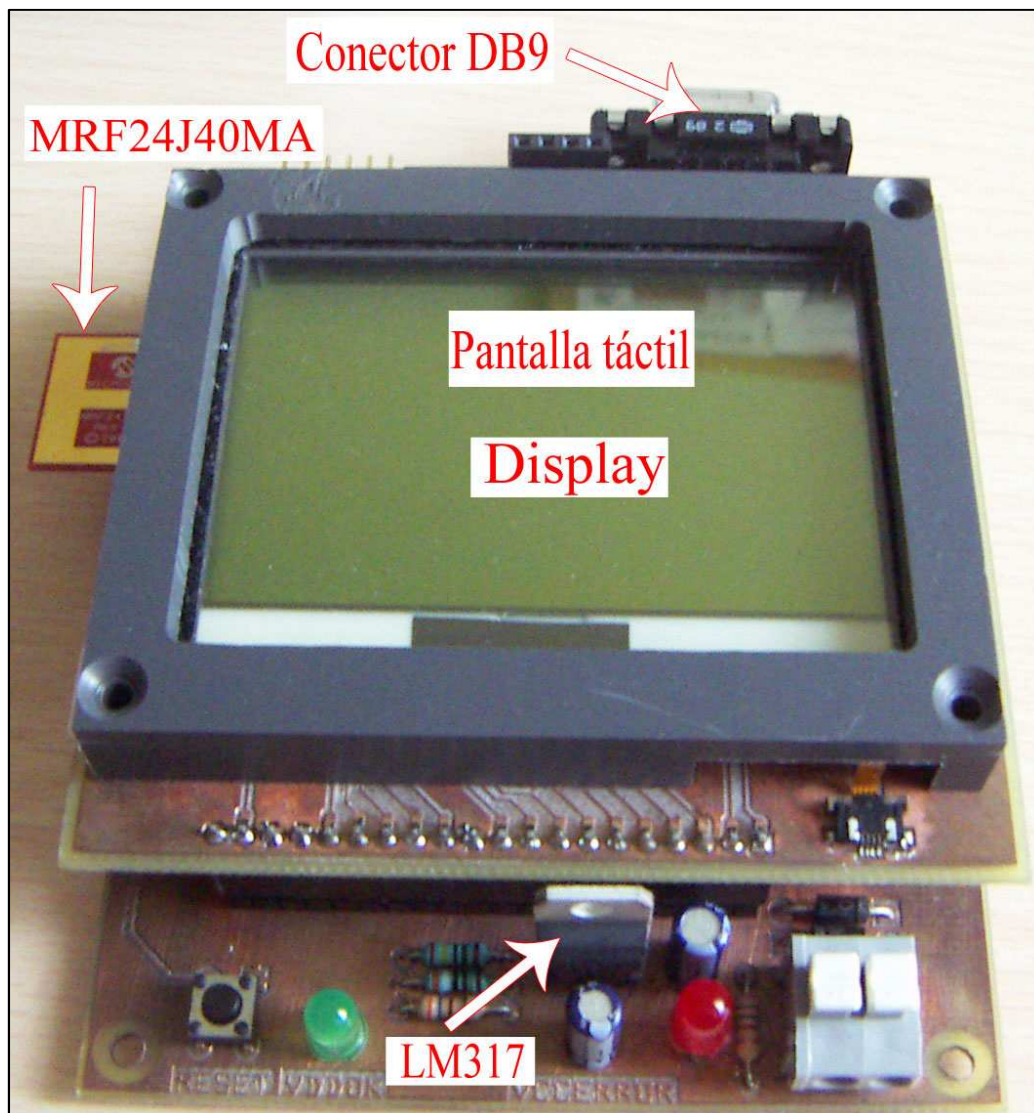


Figura 3.12 Fotografía de la placa definitiva con las pantallas conectadas

### 3.9 Conclusiones

A lo largo de este capítulo se ha detallado el funcionamiento y montaje de los diversos elementos de la placa de desarrollo. Los elementos aquí presentados han ido acoplándose de forma secuencial al microprocesador tal y como se indica en las fases explicadas durante el capítulo primero.

En primer lugar se instalaron los procesadores y transceptores de dos nodos. Con este montaje se realizaron las pruebas de configuración y funcionamiento de los PIC18F2525, los MRF24J40MA y la implementación de las comunicaciones (el protocolo de comunicaciones queda detallado en el capítulo cuarto). A continuación se ha montado el *display* a la placa temporal, programado su librería y probando su funcionamiento. Siguiendo esta línea de adición y prueba de cada elemento a la placa,

se ha añadido la pantalla táctil. Para las pruebas de RS232, se ha utilizado una placa disponible en los laboratorios del Dto. en la que las conexiones del conector DB9 y los condensadores del adaptador ya están realizados.

Tanto de forma individual a cada elemento como al sistema completo cada vez que se concluía el montaje de un componente, se han realizado pruebas de caja negra y caja blanca. Las pruebas de caja negra han consistido en depurar y seguir paso a paso el código implementado para encontrar errores o comportamientos no deseados durante la ejecución de dicho código. Las pruebas de caja blanca han servido para comprobar si a una entrada de información determinada, obteníamos como salida la respuesta esperada.

Completadas las pruebas, se ha fabricado un prototipo de la placa de desarrollo, ya con el circuito regulador de potencia. Durante la fase de pruebas del prototipo se han detectados elementos que no realizaban correctamente sus funciones debido a errores de montaje, soldaduras e interpretación incorrecta de los pines del dispositivo. Gracias a estas pruebas, se ha corregido estos contratiempos para finalmente fabricar la placa de desarrollo.

En definitiva y a la vista de lo expuesto en este capítulo, se puede dar por satisfechos los objetivos referentes a hardware expresados en el capítulo introductorio de esta memoria.



## Capítulo 4:

# Protocolo *Wireless*

El capítulo cuatro se divide en dos secciones claramente diferenciadas. Una primera parte en la que se explica las características del protocolo *wireless* elegido y las razones de dicha decisión. Esta sección acaba con una enumeración de las posibilidades de negocio de este tipo de tecnologías en el mercado.

El apartado 4.5 se encarga de desgranar la implementación en código C del protocolo descrito, detallando el funcionamiento y arquitectura de las librerías utilizadas para su explotación por la aplicación del usuario.

### 4.1 Introducción

*Microchip Wireless (MiWi)* es un protocolo de comunicaciones inalámbrico propietario de Microchip Technology Inc. (en adelante Microchip) basado en la especificación IEEE 802.15.4.



Figura 4.1 Logo de MiWi

Pese a tratarse de un protocolo privado, es de código abierto. Puede ser utilizado y modificado gratuitamente sin la compra de licencias de forma tanto privada como comercial siempre y cuando sea utilizado sobre microcontroladores y transceptores de la marca.

Siendo así, Microchip proporciona las librerías en C que implementan MiWi en el paquete *MiWi Development Environment (MiWi DE)* [17]. Este paquete está optimizado para aplicaciones de baja tasa de transmisión y bajo consumo de energía.

En la actualidad, Microchip ha desarrollado dos vertientes de MiWi. Por un lado existe *MiWi Wireless Protocol*, y por otro *MiWi P2P Wireless Protocol*. Las diferencias entre ambos son mínimas, si bien MiWi P2P se adapta mejor al entorno limitado del sistema desarrollado en este proyecto. A partir del punto 4.3, todos los datos referidos son de MiWi P2P, ya que será este el protocolo utilizado.

## 4.2 MiWi frente a Zigbee

Bajo la especificación IEEE 802.15.4 se han desarrollado, junto a MiWi, múltiples protocolos para las redes LR-WPAN. Uno de los más extendidos es Zigbee.



Figura 4.2 Logo de Zigbee

El estándar Zigbee fue desarrollado por la *Zigbee Alliance* [18], una asociación de empresas que trabaja de forma conjunta para hacer posibles productos económicos, fiables y de bajo consumo para interconectar de forma inalámbrica todo tipo de dispositivos de distintos fabricantes mediante un único protocolo compatible con los ya existentes.

Al estar basado en 802.15.4, Zigbee y MiWi comparten todas las características de la Capa Física (velocidad de transmisión, alcances, bandas de frecuencia, etc.). Es a nivel de complejidad y de licencia o certificación donde se encuentran las diferencias más significativas [19].

Microchip, pese a formar parte de la *Zigbee Alliance*, desarrolló MiWi como un protocolo más limitado pero más sencillo y con menos requisitos que Zigbee.

De esta forma, y tal y como se muestra en la Figura 4.3, la compañía implementó una Pila de Capas más sencilla, con menos capas y menos líneas de código, y por tanto, de menor tamaño. Además, al tener menos Capas, las cabeceras de las



tramas portan un mayor porcentaje de datos útiles para la aplicación, aumentando su eficiencia.

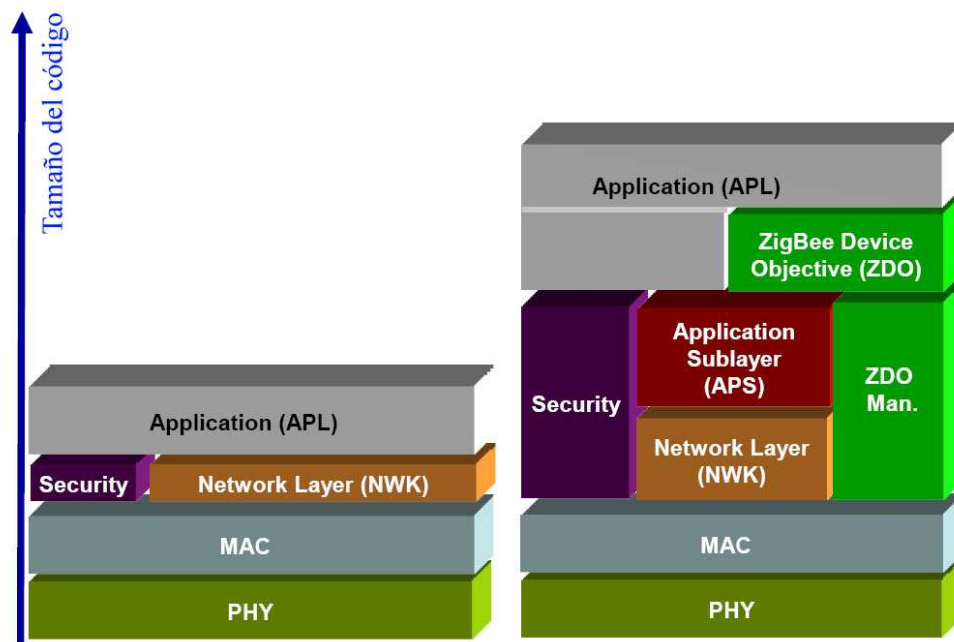


Figura 4.3 Pila de capas de MiWi (izq.) y Zigbee (drch.)

A cambio de la mayor simplicidad, las redes MiWi son notablemente de menor tamaño que una red Zigbee. Además también queda limitado el número de nodos que un mensaje puede atravesar. Otro de los grandes hándicap de la utilización de una red Zigbee es el alto precio de su uso comercial, frente al uso gratuito (siempre que se usen microcontroladores y transceptores de Microchip) de la explotación comercial de la tecnología MiWi. Sin embargo, Zigbee tiene la gran ventaja de ser un estándar consensuado por la industria, lo que garantiza la compatibilidad con otros protocolos y toda clase de hardware, independientemente de su empresa fabricante.

Característica	MiWi	Zigbee
Interoperabilidad	Propietario	Estándar de la industria
Coste comercial	Gratuito, uso de Hardware de Microchip	3.500\$/año + Cuotas de test y certificación o 9.500\$/año +Cuota de test + Certificación (solo una vez)
Tamaño del código	Coordinador <16 KB Router <16 KB Dispositivo final 2-8 KB	Coordinador 37-96 KB Router 30-64 KB Dispositivo final 2-8 KB
Red	Hasta 1024 nodos y 4 saltos	Hasta 65.536 nodos e infinitos saltos
Sobrecarga	Baja	Media

Tabla 4.1 Comparación de MiWi y Zigbee

A la vista de los datos, se ha elegido MiWi para desarrollar la parte de comunicación inalámbrica de este proyecto debido a sus reducidos requisitos de memoria y procesamiento, facilitando el uso del procesador PIC18 usado como referencia en algunas asignaturas impartidas por el Dto.

Otro dato tenido en cuenta es el tamaño reducido de la red planteada, en la que una capacidad de 1024 nodos es ya de por sí un tamaño suficientemente grande para los objetivos del proyecto.

## 4.3 Pila de Capas de MiWi

La Pila de capas de MiWi está basada en el modelo de referencia OSI visto en la Figura 2.1. Es una pila de protocolos que permite la comunicación inalámbrica entre dispositivos de forma sencilla, cumpliendo con las necesidades requeridas para los protocolos LR-WPAN.

La capa superior es la Capa de aplicación (APL). Esta capa proporciona un interfaz sencillo con la aplicación. De este modo, las capas inferiores son totalmente transparentes para la aplicación, que solo interacciona con la APL, permitiendo al programador centrarse en el desarrollo de la aplicación y no en tecnicismos del protocolo que puede no conocer.

Por debajo, está la capa de red, NWL (del inglés *Network layer*). La capa de red se construye sobre las características de la capa MAC del estándar IEEE 802.15.4, para permitir una mayor cobertura de la red. Con lo que nuevas redes podrían ser añadidas o separadas según requiera la aplicación.

Aunque no representa una capa propiamente dicha, hay que incluir una sección encargada de gestionar la seguridad de las comunicaciones en caso de que sea necesario.

A continuación, se encuentran la capa física y la capa MAC que están definidas en 802.15.4 y son comunes a todos los protocolos, aunque su implementación depende de las características del protocolo y el hardware utilizado.

### 4.3.1 Capa de aplicación

La capa de aplicación sirve como interfaz entre la aplicación y la capa de red. Es en esta capa donde se definen y configuran muchos de los parámetros que usará la capa inferior para enviar la información generada por la aplicación del usuario. Entre otras configuraciones, en este punto hay que decidir el rol que adquirirá el dispositivo, habilitar o deshabilitar los sondeos de energía y los cambios de frecuencia dinámica.

Esta capa también se encarga de mejorar la fiabilidad de la capa de red.

### **4.3.2 Capa de red**

La Capa de red implementa las necesidades del protocolo para permitir la conexión entre distintas redes, su agregación o separación. Fija el rol que tendrá cada dispositivo en la red, la topología de esta y el direccionamiento entre redes.

#### **4.3.2.1 Tipos de dispositivos**

El estándar IEEE 802.15.4 define los tipos de rol que deben existir en una red LR-WPAN. MiWi se basa en dichas especificaciones para declarar dos tipos de dispositivos en la red [20]:

##### ***Full Function Device (FFD)***

Este tipo de dispositivo tiene una funcionalidad completa. Un FFD puede ocupar el rol de coordinador PAN, iniciando la PAN (*Personal Area Network*), identificándola y controlando la explotación de la misma por otros dispositivos previamente conectados. También puede configurarse como enrutador o como dispositivo final.

##### ***Reduced Function Device (RFD)***

Un RFD opera con la mínima implementación del protocolo con el objetivo de conseguir un bajo coste y alta simplicidad. Solo pueden asociarse a un dispositivo FFD, y se inician después de que el coordinador haya establecido la red.

#### **4.3.2.2 Arquitectura de red**

La arquitectura de la red en MiWi define dos topologías de red [20].

##### ***Estrella***

Desde un punto de vista de roles. El coordinador FFD crea y regula todo el uso de la red, de tal forma que el resto de dispositivos, ya sean FFD o RFD, se conectan a él. Así, los mensajes entre dos dispositivos finales deben pasar siempre por el coordinador.

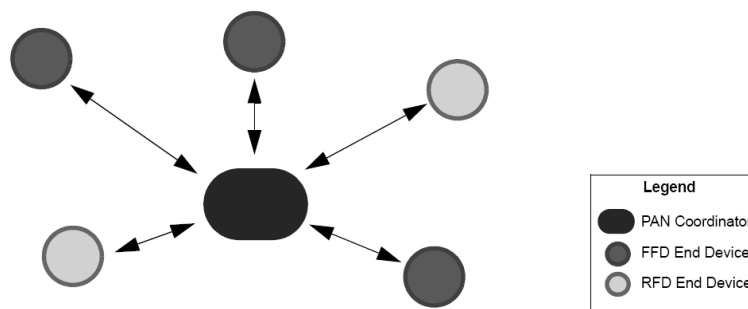


Figura 4.4 Ejemplo de topología en estrella de MiWi

Este tipo de topologías es adecuada para redes de sensores, dónde un número indeterminado de dispositivos captan información y la envían a un nodo central para ser procesada como sea conveniente.

### ***Punto a punto/peer-to-peer***

Con esta topografía, una vez que el coordinador ha creado la red, un nuevo dispositivo puede conectarse directamente con cualquier otro nodo del sistema.

Cabe destacar que un FFD puede entablar varias conexiones sin ser coordinador, pero un RFD solo puede mantener una.

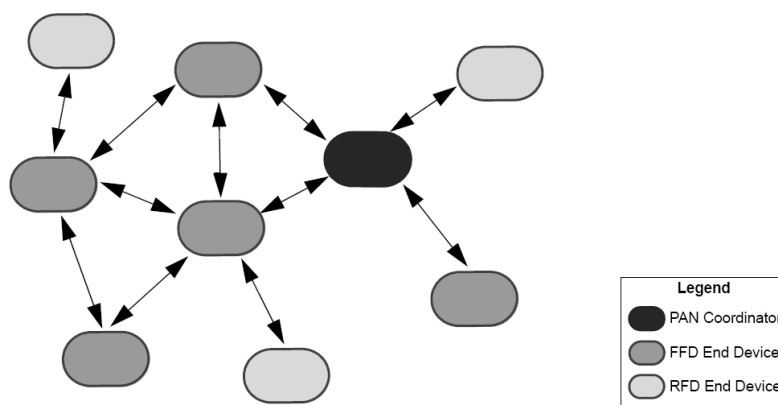


Figura 4.5 Ejemplo de topología punto a punto de MiWi

Este tipo de red es recomendado para sistemas con una alta carga de información enviada.

### ***4.3.2.3 Direccionamiento***

A nivel de red, existen dos tipos de direcciones: direcciones cortas (16 bits) y direcciones largas o direcciones IEEE (64 bits). Cada dispositivo debe tener asignada una dirección IEEE única. No puede haber dos dispositivos MiWi que posean la misma

dirección IEEE. De este modo, la dirección IEEE es asignada en el momento de la fabricación del dispositivo.

Por contra, la dirección corta es asignada por la capa de red del padre de forma dinámica cuando el nuevo dispositivo se une a una red. Dentro de una red MiWi no puede haber más de un dispositivo con igual dirección corta.

### **4.3.3 Capa MAC**

La capa MAC del protocolo provee de un interfaz entre la capa física y las capas superiores. Presenta las siguientes características.

- Asociación/disociación de dispositivos.
- Recepción y manejo de ACK.
- Mecanismos de acceso al canal.
- Validación de trama.
- Sondeo del canal.
- Control sobre las ranuras de tiempo de transmisión.

MiWi P2P utiliza el modo no-beacon con CSMA/CA no ranurado definido en 802.15.4 Este modo está diseñado para redes que necesitan transmitir poca información. Los dispositivos conectados a una red, cuando quieren transmitir, escuchan el canal para comprobar si hay alguna transmisión en marcha. En caso de que así sea, esperan un tiempo de *backoff* aleatorio antes de volver a sondear el canal. En caso de que el medio se encuentre inactivo, empieza a transmitir.

#### **4.3.3.1 Sondeo del canal**

La capa MAC tiene la capacidad de sondear el estado del canal para favorecer la calidad de los enlaces de la red. También es utilizado para reconocer otras redes existentes. Se definen dos tipos de sondeos. La duración del sondeo está limitada en segundos por el estándar 802.15.4 mediante una fórmula cerrada.

#### ***Sondeo del canal mediante Detección de Energía***

Mediante este procedimiento, la capa MAC pide un escáner de ED de la capa física para comprobar los niveles de energía de determinados canales. Con este escáner, el coordinador puede elegir el canal con menos ruido para iniciar su PAN en él.

#### ***Sondeo de canal activo***

Mediante este sondeo, un dispositivo puede comprobar si existe alguna red activa en uno o varios canales. También mide la calidad de la señal de dicha red, en caso de que exista. Este método es especialmente útil cuando el canal de operación no está predefinido por la aplicación.

#### 4.3.3.2 Cambio de frecuencia activa

MiWi provee un sistema para que una determinada PAN pueda cambiar el canal de operación sin necesidad de realizar de nuevo todo el proceso de asociación de los dispositivos. Este caso se presenta cuando se detecta una paulatina degradación del canal. Una vez elegido el nuevo canal de operación, el dispositivo que inicie el cambio de canal (no tiene por qué ser el coordinador, pero sí deberá ser FFD) debe informar al resto, haciendo una resincronización con el coordinador en caso de que sea necesario.

#### 4.3.3.3 Transmisión y recepción de tramas

Hay dos formas de enviar un paquete: *broadcast* o *unicast*. Los paquetes *broadcast* se envían a todos los dispositivos de la red, mientras que los mensajes *unicast* tienen una dirección única de destino. La primera clase no requiere confirmación de recepción mediante ACK, pero sí el segundo tipo.

Como método de ahorro de energía, los nodos pueden estar en modo *sleep*. Cuando el dispositivo despierta, realiza una petición de datos al coordinador para conocer si existen mensajes para él.

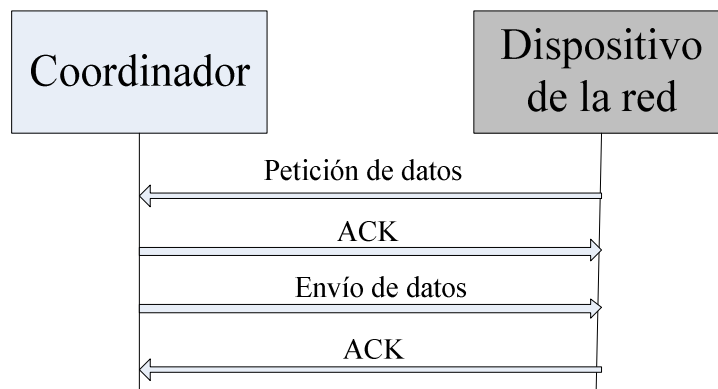


Figura 4.6 Tráfico de mensajes bajo petición

Si es el dispositivo cuando necesita enviar información, compete por el medio mediante CSMA/CA no ranurado. Cuando consigue el uso del canal, envía los datos y se mantiene a la espera del ACK en caso de que se trate de un paquete unicast, tal y como se muestra en la Figura 4.7.

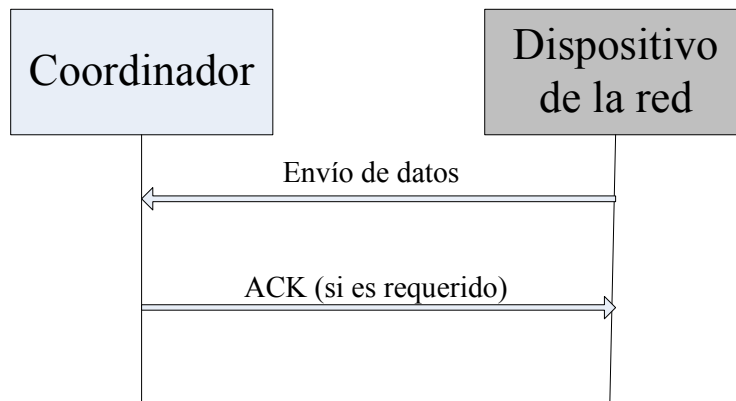


Figura 4.7 Tráfico de mensajes nodo-coordinador

#### 4.3.3.4 Creación de una red

Un FFD es el único que puede crear una nueva red. Después de determinar un canal de operación, ya sea de forma predefinida por la aplicación como después de realizar sondeos, y tomar un identificador de red (PAN ID), el FFD espera a recibir una petición de asociación.

##### **Asociación**

Un dispositivo que quiera unirse a una determinada red, envía una petición de asociación mediante un mensaje *broadcast*. Todos los FFD responderán a dicha petición con un mensaje de aceptación de petición siempre que tengan los recursos suficientes.

Si el nuevo dispositivo es un FFD, podrá establecer tantas conexiones como la aplicación requiera y siempre y cuando tenga los recursos necesarios disponibles. Si es un RFD, únicamente establecerá conexión con la primera confirmación que reciba.

##### **Disociación**

El dispositivo, ya sea un coordinador queriendo disociar a otro nodo o un nodo queriendo disociarse de una red, envía un mensaje de desconexión al coordinador/dispositivo. Recibida la notificación, el nodo envía un ACK confirmando la recepción.

#### 4.3.4 Capa física

La capa física es la responsable de la transmisión y la recepción de datos en un canal de radio y acorde con las técnicas de modulación y ensanchamiento. El IEEE 802.15.4 ofrece tres bandas de frecuencia en las cuales operar: 2.4 GHz, 915 MHz y 868 MHz.

La frecuencia de 2.4 GHz, especifica la operación en la banda industrial, médica y científica (ISM), la frecuencia de 868 MHz opera únicamente en Europa y 915 MHz en Estados Unidos. El estándar IEEE 802.15.4 utiliza la técnica DSSS (*Direct Sequence Spread Spectrum*) para transmitir la información a través del medio. La disponibilidad internacional de la banda de los 2.4 GHz ofrece ventajas en términos de mercados más amplios, por otro lado las bandas de 868 MHz y 915 MHz ofrecen una alternativa a la congestión creciente e interferencias asociadas a una banda pública. Las velocidades de transmisión son de 250 kbps en la banda de 2.4 GHz, 40 kbps en la banda de 915 MHz y 20 kbps en la banda de 868 MHz.

En el caso de la placa de desarrollo, el transceptor MRF24J40MA únicamente funciona a 2.4 GHz.

#### **4.3.4.1 Características generales**

##### ***Modos del transceptor***

El transceptor tiene tres modos de funcionamiento: transmisión, recepción y *sleeping* (descanso). Bajo petición de la capa MAC la capa física debe conmutar entre estos tres estados. El estándar exige que la conmutación entre transmisión y recepción, o viceversa, se haga en menos de 12 símbolos.

##### ***Detección de energía (ED) en el canal***

Es una estimación de la energía presente en un canal, siendo ese valor analizado con respecto a un valor umbral predeterminado (umbral ED). Esta medida es usada para la selección del canal, y por CCA (*Clear Channel Assessment*) para determinar si el canal está libre u ocupado.

##### ***Indicador de calidad del enlace (LQI)***

LQI (*Link Quality Indication*) indica la medida de fuerza/calidad del paquete recibido, esta medida puede ser implementada usando detección de energía (ED).

##### ***Prueba de Clear Channel (CCA)***

CCA (*Clear Channel Assessment*). Esta operación es responsable de informar el estado de actividad en el medio (libre u ocupado). El CCA tiene tres modos de operación:

- Modo de detección de energía: el CCA informa que el canal está ocupado si el valor de la energía detectada está sobre el umbral ED.
- Modo de sondeo de portadora: el CCA notifica que el canal está ocupado solamente si detecta una señal con las técnicas de modulación y ensanchamiento establecidas por el estándar 802.15.4, sin importar que dicha señal esté por encima o por debajo del umbral ED.
- Modo de sondeo de portadora con detección de energía: combinación de los dos modos anteriores.



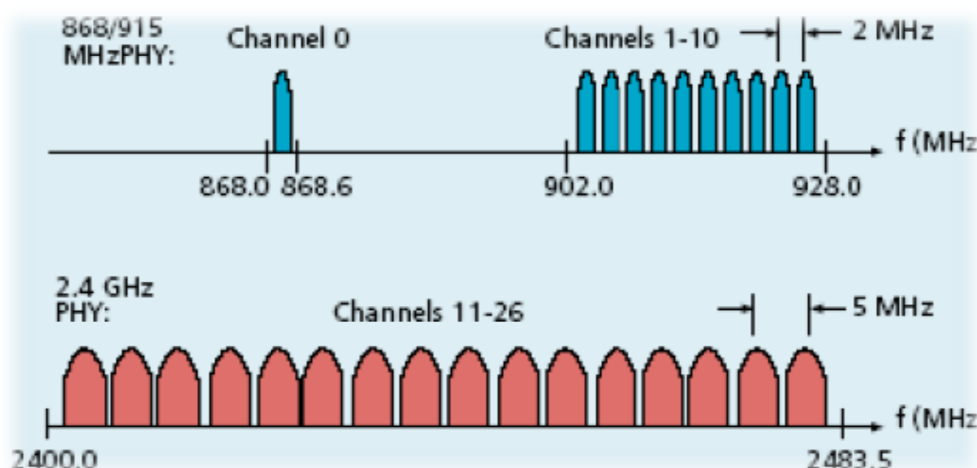
### ***Selección de la frecuencia del canal***

La IEEE 802.15.4 define 27 canales diferentes; por lo tanto, la capa física debe sintonizar al dispositivo dentro del canal a utilizarse.

#### ***4.3.4.2 Canalización de las bandas de frecuencia***

IEEE 802.15.4 define 27 canales de frecuencia repartidos de forma dispar entre las tres bandas de operación.

La banda 868 MHz soporta un solo canal; la banda 915 MHz soporta diez canales entre los 902.0 y 928.0 MHz, equiespaciados cada 2 MHz; la banda de 2.4 GHz soporta 16 canales entre 2.4 y 2.4835 GHz, con un espacio entre frecuencias centrales de 5 MHz con el objetivo de facilitar los requerimientos de filtrado en la transmisión y en la recepción.



*Figura 4.8 Estructura de canalización de 802.15.4 [5]*

El estándar fue diseñado para implementar una selección dinámica de canales, a través de diversas funcionalidades implementadas en la capa de enlace. La capa MAC incluye funciones de búsqueda mientras que la PHY contiene varias funciones de bajo nivel, tales como la detección de los niveles de energía recibidos, indicadores de calidad en el enlace así como de conmutación de canales, lo que permite asignación de canales y agilidad en la selección de frecuencias.

#### ***4.3.4.3 Modulación***

MiWi utiliza dos tipos de modulación en función de la banda de operación. En las bandas de 868 y 915 MHz se usa BPSK, mientras que para la banda ISM, se modula mediante O-QPSK.

En comparación, la modulación BPSK permite una sensibilidad de entre 6 y 8 dB mayor con respecto a O-QPSK, sin embargo, requiere una eficiencia en términos de

energía por bit de, aproximadamente, 2 dB por encima de la modulación usada en la banda de 2.4 GHz.

#### 4.3.4.4 Sensibilidad, potencia e interferencias

Las especificaciones de sensibilidad de IEEE 802.15.4 especifican -85 dBm para la frecuencia de 2.4 GHz y de -92 dBm para las frecuencias de 868 y 915 MHz.

Dichos valores incluyen suficiente margen para las tolerancias que se requieren debido a las imperfecciones en la fabricación, de la misma manera que permite implementar aplicaciones de bajo presupuesto.

El estándar IEEE 802.15.4 especifica que cada dispositivo debe ser capaz de transmitir al menos a 1 mW, pero dependiendo de las necesidades de la aplicación, la potencia de transmisión puede ser mayor o menor.

Los dispositivos típicos (1mW) se espera que cubran un rango de entre 10 y 75 m; sin embargo, con una buena sensibilidad y un incremento moderado en la potencia de transmisión, se obtiene mayores coberturas.

#### 4.3.5 Formato de mensaje

Las cabeceras de un mensaje MiWi P2P se han definido como un subconjunto de las opciones ofrecidas por el estándar 802.15.4. En la Figura 4.9 se puede ver dicho formato de un mensaje generado por las Capas de red y MAC antes de ser transferido a la Capa física [20].

Control de trama 2 bytes	Número de secuencia 1 byte	PAN ID destino 2 bytes	Dirección destino 2/8 bytes	PAN ID fuente 0/2 bytes	Dirección fuente 8 bytes	Payload variable	Secuencia de comprobación de trama 2 bytes
-----------------------------	-------------------------------	---------------------------	--------------------------------	----------------------------	-----------------------------	------------------	--

Figura 4.9 Estructura del mensaje de red

- Control de trama: estos dos bytes se encargan de: indicar el tipo de trama, indicar si el paquete se ha encriptado, si el ACK es requerido, si el destino está en la misma red y el modo de direccionamiento usado.
- Número de secuencia: el número de secuencia lleva el conteo del número de trama enviado.
- PAN ID destino/fuente: identificador de red de la Pan de destino/origen del mensaje.
- Dirección destino/fuente: dirección del dispositivo de destino/origen. La dirección de fuente es siempre la dirección larga.
- *Payload*: datos recibidos desde la capa de aplicación.
- Secuencia de comprobación de trama: son los 2 últimos byte del mensaje y sirven para comprobar que no se han producido errores durante la transmisión física del mensaje.

Antes de la transmisión, la Capa física introduce un encabezado de sincronización (SHR) y un encabezado de longitud de paquete.

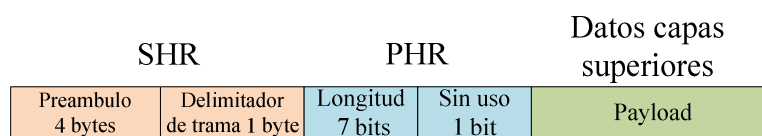


Figura 4.10 Estructura del paquete de capa física

- *Synchronization Header (SHR)*: este campo se utiliza para la correcta sincronización de la trama.
- *PHY Service Data Unit (PSDU)*: en el *payload* se encapsula la trama recibida de la capa MAC, cuya longitud máxima es de 127 bytes, haciendo un máximo total por paquete de 133 bytes, con una duración máxima de 4.25 ms para la banda de 2.4 GHz, 26.6 ms para la banda de 915 MHz y 53.2 ms para la banda de 868 MHz.
- *PHY Header (PHR)*: declara el largo de la PSDU.
- *Payload*: datos procedentes de las capas superiores.

#### 4.3.6 Seguridad

Desde el punto de vista de la seguridad, las redes inalámbricas son altamente vulnerables, debido a que no se requiere tener acceso a un medio cableado para participar en las comunicaciones, por lo que cualquier oyente no deseado puede captar la información enviada. Además, los dispositivos que trabajan en redes LR-WPAN se enfrentan a una limitación computacional evidente. Esto quiere decir que los sistemas de seguridad son más difíciles de implementar en sistemas que trabajan bajo 802.15.4. A la hora de implementar los mecanismos de seguridad, se ha tenido en cuenta que los dispositivos pueden no tener acceso a una base de datos. De esta forma, los distintos dispositivos deben ser capaces de gestionar la seguridad por sí mismos. Por último, está orientado hacia el envío de pequeños paquetes de información, por lo que los mecanismos de seguridad no deberían aumentar en exceso el tamaño de la cabecera.

Con estas premisas, IEEE 802.15.4 define mecanismos de seguridad para las capas MAC, de red y de aplicación. El mecanismo criptográfico utilizado se basa en el uso de claves simétricas. La clave en cuestión debe ser generada por las capas superiores. El mecanismo criptográfico debe ser capaz de asegurar:

- Confidencialidad de los datos, es decir, que la información enviada solo ha de llegar a los dispositivos adecuados.
- Autenticación de los datos (provengan de una fuente válida).
- Detección de datos duplicados.

De acuerdo al estándar, la clave utilizada puede ser conocida por los dos dispositivos que participan en la comunicación o por un grupo de dispositivos. Si utilizamos una única clave para un grupo de dispositivos ganamos en simplicidad pero, por contra, estamos desprotegidos ante el ataque de un dispositivo de la propia red.

Microchip utiliza cifrado mediante la técnica XTEA (*eXtended Tiny Encryption Algorithm*) [21]. XTEA es un cifrado de bloques de 64 bits con claves de 128 bits, basado en el cifrado de Feistel [22]. El algoritmo requiere un consumo de memoria muy limitado, y pudiendo variar el número de rondas, se puede ajustar el tiempo de ejecución a los requerimientos de la aplicación (aunque, a menor número de rondas, más fácil de romper la clave).

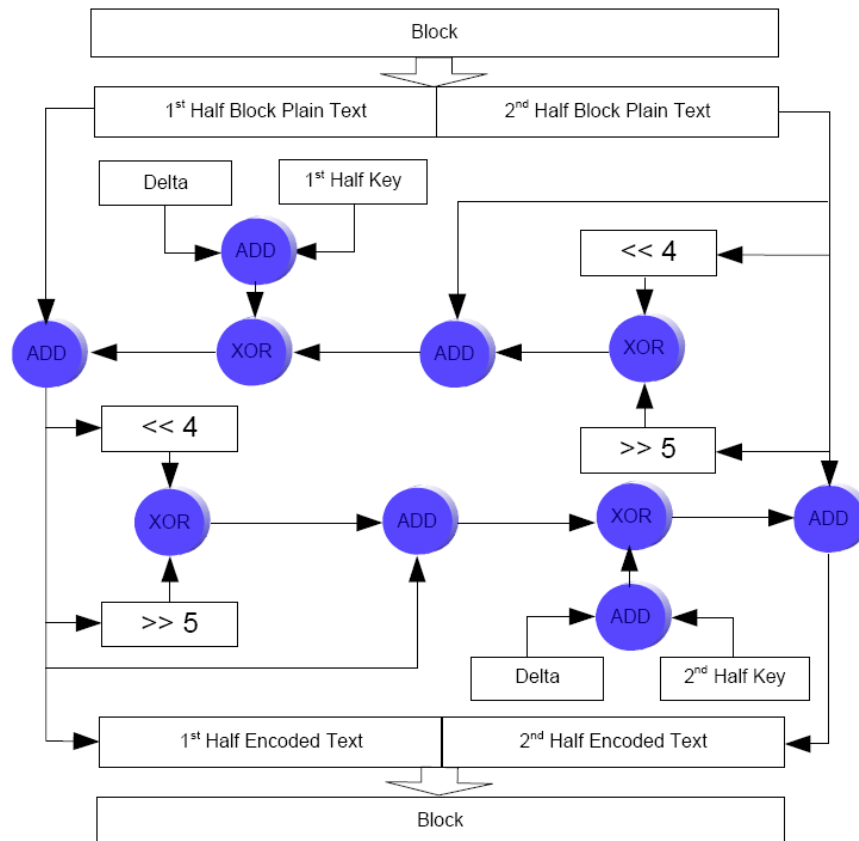


Figura 4.11 Diagrama de funcionamiento de XTEA

En MiWi, además del uso del sistema XTEA para implementar la seguridad de la información, se han diseñado distintos modos de operación.

El más sencillo es el modo ECB. Este modo divide el mensaje en varios bloques de mismo tamaño, y luego aplica la codificación a cada uno de ellos. Este modo no oculta el patrón de datos (para misma entrada, siempre dará la misma salida).

El modo *Counter* (CTR) elimina este inconveniente mediante un contador que impide que se repita el número de rondas que aplica el cifrado, pero a costa de aumentar el tiempo de procesamiento o los recursos utilizados en el cifrado. Sin embargo, el cifrado en modo CTR puede no resultar eficiente para evitar ataques externos.

Aparte del modo CTR, MiWi define los modos de operación para autenticar el mensaje. El modo de autenticación estándar es *Cipher Block Chaining Message Authentication Code*, CBC-MAC. CBC-MAC es utilizado para autenticar el mensaje, pero por si mismo no realiza cifrado.

El modo CCM (*Counter with CBC-MAC*) es el más completo de todos, ya que realiza CBC-MAC seguido por la aplicación de cifrado CTR.

## 4.4 Mercado y aplicaciones

En general, MiWi es idóneo para redes estáticas, escalables y un número no excesivamente alto de dispositivos, pocos requisitos de ancho de banda o calidad de servicio, uso frecuente y duración de la batería lo más longeva posible. En otras condiciones, según los factores que cambien, puede ser una buena alternativa usar otros protocolos dentro de la familia 802.15 como *Bluetooth* o *Zigbee*, u otras tecnologías distintas y más potentes como *Wi-Fi* o *UMTS*.

Algunas de las aplicaciones comerciales podrían ser:

- Control remoto de otros equipos en lugar de RF o Infrarrojos.
- Iluminación: encendido, apagado, control de intensidad.
- Control mediante sensores: temperatura, presión, humedad, nivel, humo, etc.
- Control de acceso.
- Regadío.
- Ventilación.
- Manejo de carga de trabajo
- Diagnostico de equipos.
- Teleseguimiento de pacientes
- Periféricos de PC y videoconsola: mandos, ratón, teclado, impresora, etc.
- Identificación de vehículos.
- Redes de sensores contra incendios.
- Detectores de actividad sísmica.

## 4.5 Diseño e implementación de MiWi en la placa de desarrollo

Microchip proporciona dentro de su paquete de librerías descargable la implementación en C de su protocolo MiWi, incluyendo proyectos de ejemplo realizados con el entorno de desarrollo de la marca. Para realizar este proyecto, se han utilizado estas librerías como base a partir de las cuales se ha ido ajustando el código a los requerimientos planteados y al hardware que específicamente se ha desarrollado.

MiWi DE ha sido diseñado para dar cobertura con una sola librería a las distintas modalidades del protocolo MiWi, así como al uso de distinto hardware, pudiendo elegir entre diversos transceptores y microprocesadores o incluso elegir usar algún tipo de placa integrada de demostración que tiene la marca puesta a la venta (por ejemplo, la placa *PicDem Z*) sin tener que realizar modificación alguna en el núcleo de la Pila

MiWi, únicamente hay que realizar modificaciones, con mayor o menor grado de dificultad y número, en la configuración.

Para hacer esto posible, se definen dos planos completamente diferenciados, uno de configuración y otro de operación. También se define una interfaz entre la aplicación y el protocolo de comunicaciones (MiApp) y otra entre dicho protocolo y la Capa MAC y la Capa física (MiMAC). Esta composición de la librería se refleja en la Figura 4.12 y se explica en los subapartados siguientes.

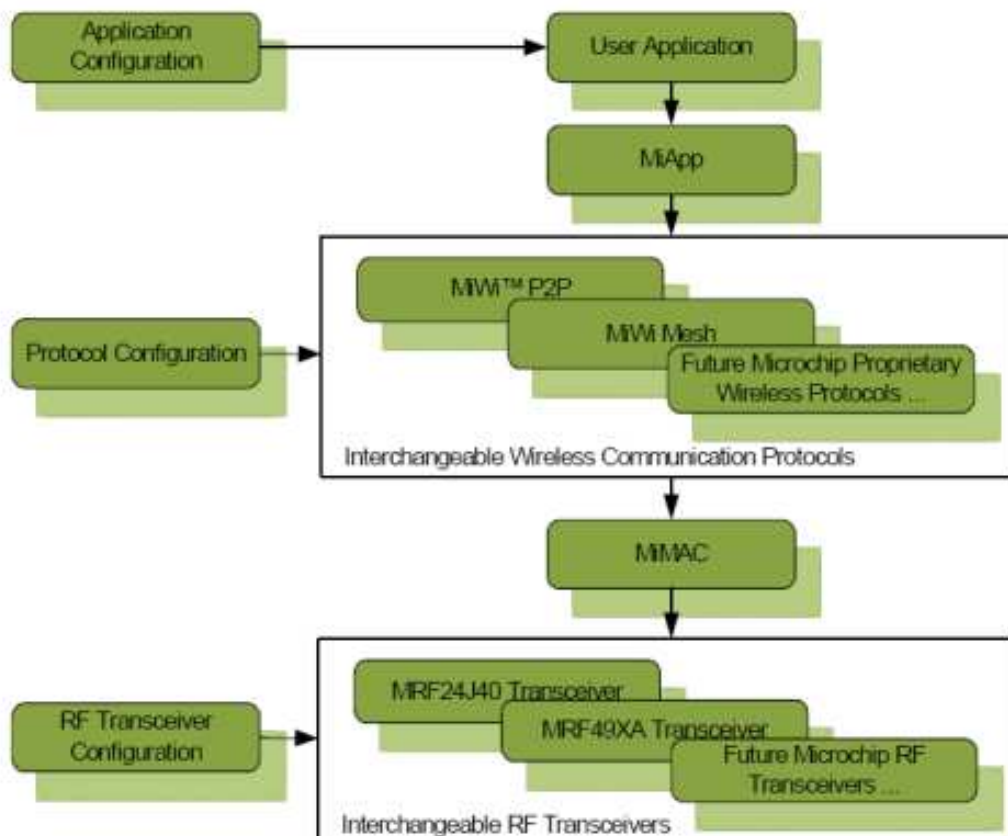


Figura 4.12 Diagrama de bloques de MiWi DE [23]

#### 4.5.1 Nivel de aplicación

La aplicación es el software que usará las comunicaciones de la forma que el usuario requiera. Además, es la que otorga al sistema de una funcionalidad determinada.

En este punto de configuración quedan definidos inequívocamente el modelo de MiWi y el transceptor que se va a utilizar. Además de otros parámetros indispensables para el correcto funcionamiento de las comunicaciones, como son el rol del dispositivo (coordinador o no), el tamaño de los *buffers* de entrada y salida, uso de mensajes broadcast, etc. También se realiza la configuración del canal a utilizar, predeterminándolo desde la aplicación, o activando el sondeo de canal de MiWi.

En la siguiente figura se muestran algunos de los parámetros de configuración mencionados en el párrafo anterior. El código completo de configuración está en el archivo ConfigApp.h dentro de la carpeta de cada Nodo, en el anexo electrónico.

```
//-----
// Definición del Protocolo: MiWi P2P Stack.
//-----
#define PROTOCOL_P2P

/*****/
// NWK_ROLE_COORDINATOR especifica el tipo de dispositivo que es, final
// o coordinador. En MiWi P2P no es necesario
/*****/
#define NWK_ROLE_COORDINATOR

/*****/
// Definición del transceptor
/*****/
#define MRF24J40
#define IEEE_802_15_4

/*****/
// TX_BUFFER_SIZE define el tamaño máximo del Payload de la aplicación
// al transmitir
/*****/
#define TX_BUFFER_SIZE 127

/*****/
// RX_BUFFER_SIZE define el tamaño máximo del Payload de la aplicación
// al recibir
/*****/
#define RX_BUFFER_SIZE 127

/*****/
// MY_PAN_ID define el identificador de PAN
/*****/
#define MY_PAN_ID          0x1234

/*****/
// P2P_CONNECTION_SIZE define el número máximo de conexiones P2P a la
// vez
/*****/
#define CONNECTION_SIZE    4

/*****/
// ENABLE_ED_SCAN habilita al dispositivo realizar barridos de energía
// para encontrar el canal con menor ruido.
/*****/
#define ENABLE_ED_SCAN

/*****/
// ENABLE_SECURITY activa la encriptación de los mensajes
/*****/
#define ENABLE_SECURITY
```

*Figura 4.13 Ejemplo de configuración de aplicación*

## 4.5.2 MiApp

MiApp [23] es la interfaz de programación desarrollada por Microchip para hacer lo más transparente posible al usuario la utilización del protocolo inalámbrico elegido.

Con esta premisa en mente, la aplicación debe incluir los ficheros en los que se implementa esta interfaz, donde están definidas las funciones que el usuario podrá usar para explotar la red de la forma que considere conveniente. Las funciones más importantes programadas en MiApp se encuentran en la tabla siguiente, junto con una breve descripción de su cometido.

Función	Descripción
DiscardMessage	Descarta el mensaje recibido para poder procesar el siguiente.
EstablishConnection	Establece una conexión con uno o más dispositivos
MessageAvailable	Comprueba si se ha recibido un mensaje
MiApp_BroadcastMessage	Envía un mensaje <i>broadcast</i>
MiApp_ConnectionMode	Selecciona uno de los modos disponibles de conexión
MiApp_FlushTx	Reinicia el puntero del buffer de transmisión
MiApp_ProtocolInit	Inicializa el transceptor y el <i>stack</i> MiWi
MiApp_RemoveConnection	Elimina una conexión previamente establecida
MiApp_SetChannel	Define el canal de operación de la red.
MiApp_UnicastAddress	Envía un mensaje <i>unicast</i> a un nodo mediante su dirección
MiApp_UnicastConnection	Envía un mensaje <i>unicast</i> a un dispositivo de la tabla de conexiones
MiApp_WriteData	Encola un byte en el buffer de transmisión
SearchConnection	Inicia la búsqueda de nodos existentes en la vecindad. Es el <i>Active Scan</i>
StartConnection	Habilita al nodo para establecer conexiones

Tabla 4.2 Funciones más importantes de MiApp

Las funciones relacionadas con el inicio de la red y el envío y recepción de mensajes son las más utilizadas por la aplicación.

### 4.5.3 Nivel de MiWi P2P

La implementación del protocolo realizada en MiWi DE está programada en archivos de código C independientes que interaccionan con las interfaces MiApp y MiMAC. Este aislamiento proporciona transparencia para la aplicación del usuario a la hora de usar las comunicaciones.

La aplicación deberá ocuparse únicamente de configurar las opciones para amoldar el protocolo a los requisitos y al funcionamiento deseado. Algunos de estos aspectos configurables se muestran en el siguiente extracto de código.



```

/*****
// RFD_DATA_WAIT es el Timeout definido entre la petición de mensaje y
// su desechamiento
*****/
#define RFD_DATA_WAIT                0x00003FFF

/*****
// CONNECTION_RETRY_TIMES es el número máximo de veces que el nodo inten-
// tara reestablecer una conexión. Posteriormente el control retornará a
// la aplicación.
*****/
#define CONNECTION_RETRY_TIMES        3

/*****
// CONNECTION_INTERVAL define el intervalo, en segundos, entre dos
// peticiones de conexión.
*****/
#define CONNECTION_INTERVAL            2

/*****
// FA_BROADCAST_TIME define el número de veces que emite en broad cast
// el nuevo cana después de realizar el salto de frecuencia.
*****/
#define FA_BROADCAST_TIME              0x03

/*****
// RESYNC_TIMES define el número máximo de veces que se intenta resincro-
// nizar en todo los canales antes de devolver el control a la aplicación.
*****/
#define RESYNC_TIMES                    0x03

/*****
// ACTIVE_SCAN_RESULT_SIZE define el máximo resultado que puede esperar
// tras el escaner activo. Si el resultado superar este límite, se descar-
// tará
*****/
#define ACTIVE_SCAN_RESULT_SIZE        4

```

*Figura 4.14 Ejemplo de configuración de MiWi P2P*

#### 4.5.4 MiMAC

MiMAC es la interfaz de programación entre el protocolo de comunicaciones y la Capa física. Además de permitir la conexión de distintos transceptores sin realizar modificaciones en los niveles superiores, MiMAC se encarga e implementa todos los trabajos realizados por la Capa MAC. También es la encargada de realizar la encriptación/desencryptación de los mensajes en caso de que la aplicación lo solicite.

La gran ventaja de usar MiMAC, es el hecho de que este interfaz se encarga de la gestión y control del transceptor. Gracias a eso el usuario puede olvidarse de tener que lidiar con los registros y el funcionamiento de un determinado transceptor.

Aunque la aplicación no suele hacer uso de las funciones de MiMAC, las más importantes se muestran en la Tabla 4.3.

Función	Descripción
MiMAC_DiscardPacket	Notifica a la capa MAC que el paquete actual ha sido procesado y puede ser eliminado
MiMAC_Init	Inicia el comportamiento de la interfaz MiMAC y el transceptor
MiMAC_ReceivedPacket	Indica si existe un mensaje sin procesar
MiMAC_SendPacket	Envía un paquete
MiMAC_SetPower	Modifica la potencia de salida del transceptor. Otorga a la aplicación la capacidad de variar la potencia de emisión de forma dinámica

*Tabla 4.3 Funciones más importantes de MiMAC*

#### 4.5.5 Nivel del Transceptor MRF24J40

MiWi DE provee del código necesario para elegir y utilizar diversos transceptores, entre los que se encuentra el MRF24J40. De esta forma, al igual que con los elementos anteriores, el usuario puede programar su aplicación y usar el transceptor de forma transparente gracias al uso de MiApp, MiMAC y las configuraciones adecuadas.

## 4.6 Conclusiones

A la vista de los apartados anteriores de este capítulo, se puede decir que MiWi P2P es un protocolo de comunicaciones adecuado para redes pequeñas y de baja utilización de los enlaces.

Debido a la alta modularización de la librería, se puede adaptar el software para cubrir todas las necesidades que el usuario pretenda satisfacer utilizando esta tecnología. Modificando los valores de configuración del protocolo, o incluso cambiando las definiciones de pines, se puede amoldar el mismo software a otros entornos, tanto de aplicación como hardware, sin tener que realizar grandes cambios tanto en la librería como en el código de ejecución de la aplicación.

Para facilitar el seguimiento del código de la librería se han traducido al castellano los comentarios de los archivos de código.

Finalmente, en el anexo B se encuentra una guía rápida de configuración de MPLAB IDE para poder compilar el software del proyecto y poder cargarlo en los microprocesadores.

## **Capítulo 5:**

# **Ejemplo de uso: Monitorización de Temperaturas**

En este capítulo se describe un ejemplo de aplicación a modo de demostración de uso de la placa desarrollada. En primer lugar se plantea el entorno del ejemplo y sus características de forma general; posteriormente, se enfoca sobre los distintos dispositivos del sistema en dos apartados diferenciados, uno para el dispositivo cuyo rol será el de coordinador de la red y otro para los dispositivos que adoptarán el papel de nodo satélite.

### **5.1 Introducción**

Se pretende simular un sistema distribuido de monitorización de temperaturas, para ello se plantea un escenario de demostración con tres dispositivos diferentes: un coordinador FFD y dos nodos satélite RFD.

El coordinador estará cargado con una aplicación que gestionará el uso de la red y usará todos los elementos del hardware diseñado y será a través del cual el usuario realizará todas las interacciones con el sistema. Mientras que los nodos RFD únicamente utilizarán el microprocesador, el transceptor y un sensor de temperatura digital, en este caso, un LM71.

De forma global, y una vez establecida la conexión de ambos nodos RFD al coordinador, el usuario podrá solicitar a dichos nodos (a través de la pantalla táctil) la

captura de la temperatura actual del lugar donde se encuentre. Al recibir la petición, el nodo satélite usará el sensor para capturar dicho valor. Tras ajustar a la resolución deseada de un solo byte, se envía la información al coordinador. Será el coordinador el encargado de convertir el byte a información representable por el LCD y el USART.

## 5.2 Incorporación del sensor LM71 al sistema HW

El LM71 [24] es un sensor de temperatura de alta precisión y baja potencia. Para este proyecto se ha utilizado el modelo de 5 pines conectado al microprocesador mediante una conexión SPI.

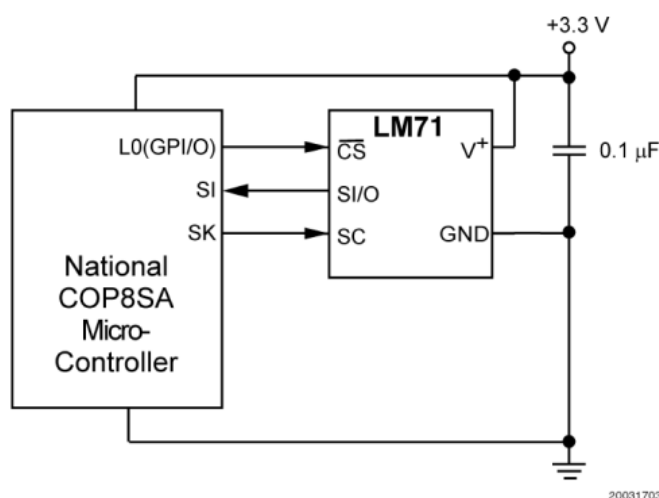


Figura 5.1 Configuración típica del LM71 5 pines [24]

Este sensor tiene un rango de temperaturas de  $-40^{\circ}\text{C}$  a  $150^{\circ}\text{C}$ , utilizando los 14 bits más significativos de los 2 bytes de información que usa (los 2 bits restante siempre valen 1). Para el software de demostración del prototipo, se ha aumentado la escala del sensor, pasándose a utilizar únicamente 1 byte de información. Con esta medida, se pierden las temperaturas negativas y la escala pasa a ser de  $0.25^{\circ}$  por bit. Pese a perder prestaciones, sigue siendo perfectamente válido para el objetivo que pretende cumplir este dispositivo.

Previamente a la implementación de la función que obtiene la medición del sensor, hay que definir que pin del microprocesador usará la aplicación como *Chip Select* para activar o desactivar la recogida de datos del LM71. En el supuesto aquí implementado, se ha utilizado el pin 7 RA5 para el manejo del *Chip Select*. Por otro lado, el sensor también hará uso de los pines 14 SCL y 15 SDI para señal de reloj y puerto de entrada de la comunicación SPI.

La función utilizada para obtener la temperatura a través del LM71 está disponible en el anexo electrónico de esta memoria, concretamente en los archivos `Nodo_2.c` y `Nodo_3.c`, siendo idéntica en ambos.

## 5.3 Descripción de las funcionalidades implementadas en los dispositivos

Para ver el software completo relacionado con el coordinador, los nodos con el sensor de temperatura, el procesamiento de paquetes y la rutina de interrupción, se debe ir al código incluido en el anexo electrónico.

### 5.3.1 Coordinador

El Coordinador es el dispositivo que ha de arrancar en primer lugar. El primer cometido de la aplicación que se ha programado es configurar e inicializar los dispositivos y módulos que van a ser utilizados: configuración del PIC18F2525 y sus puertos, reset e inicio del LCD, configuración del Conversor A/D y del módulo USART. A continuación, se establecen la red LR-WPAN con todas sus características y se espera a recibir alguna petición de conexión.

Una vez que uno o más nodos satélites han establecido conexión con el coordinador, el usuario podrá solicitarles el envío de un nuevo valor de la temperatura ambiental a través de pulsaciones en la pantalla táctil y navegando por un sencillo sistema de menús. El diagrama de flujo del SW del coordinador FFD se muestra en la Figura 5.2.

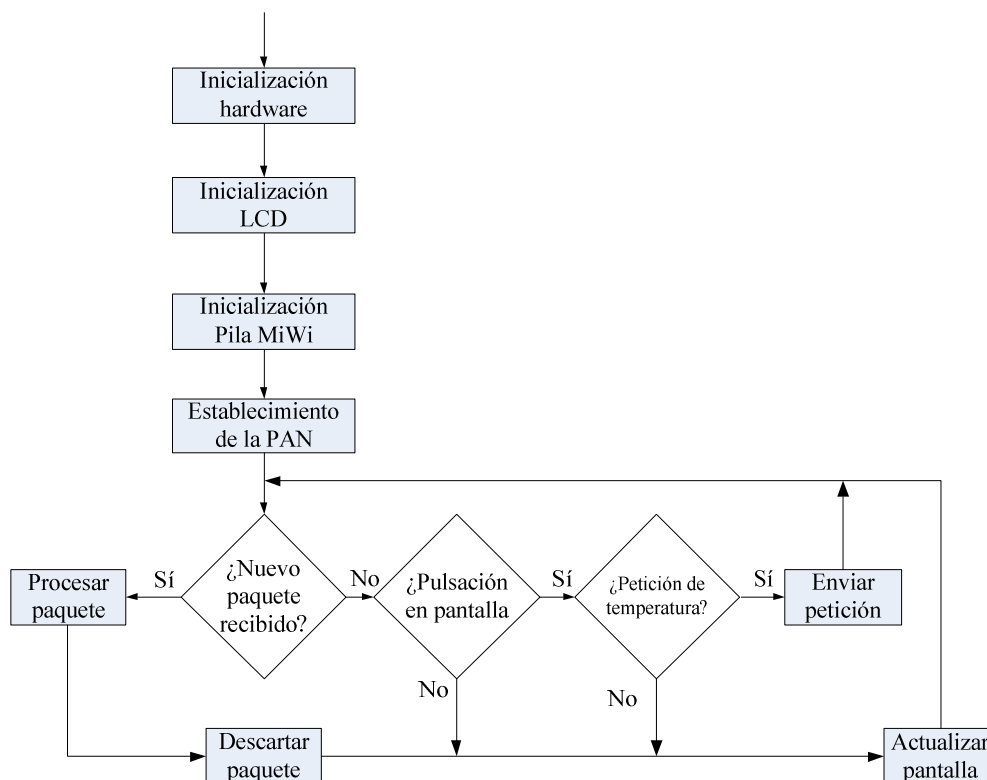
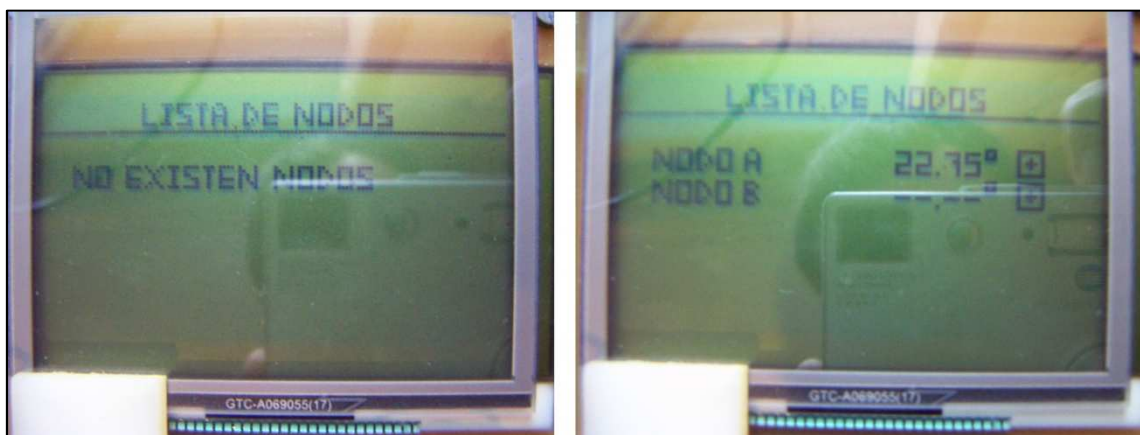


Figura 5.2 Diagrama de flujo de la aplicación del coordinador

El interfaz del usuario con el sistema se gestiona a través del *display* gráfico y de la pantalla táctil montada sobre él. Se ha programado un sistema de menús compuesto por dos tipos de pantallas: una muestra al usuario los nodos conectados, desde ninguno hasta un máximo de cuatro, y su última temperatura recibida; la segunda pantalla se accede pulsando un botón a la derecha de cada nodo, y muestra la última temperatura recibida, un botón de solicitud de nueva temperatura, y otro botón para volver a la vista general.

En la Figura 5.3 se muestran dos fotografías de la pantalla del nodo coordinador en la que se puede ver el menú inicial. La fotografía de la izquierda está tomada cuando no se ha conectado nodo alguno, mientras que la de la derecha muestra dos nodos conectados, habiendo recibido ya una temperatura de uno de ellos.



*Figura 5.3 Pantallas del sistema de menús: vista general*

En la figura siguiente, las fotografías se han tomado cuando se mostraba por pantalla el menú del Nodo A (a la izquierda) y del Nodo B (a la derecha).



*Figura 5.4 Pantallas del sistema de menús: vista de un nodo*

### 5.3.2 Procesado de paquetes recibidos por el coordinador

Dado que el *payload* de los mensajes recibidos es de un solo byte, la mayor dificultad del procesamiento del mensaje reside en transformar ese byte que representa la temperatura a un tipo de dato que pueda ser representado por el LCD y enviado por el USART. Esta transformación la realiza la función *ConversorBitMap*, que recibe por parámetros el byte a transformar y el nodo del que procede.

Como se puede apreciar en las figuras 5.3 y 5.4, el formato de la temperatura será de 4 dígitos, 2 para la parte entera y 2 para la parte decimal (por ejemplo, 24. 25). Para identificar las decenas, la función elimina los 2 bits menos significativos, quedando únicamente la parte entera. De este modo se obtiene un número binario entero, y se extrae fácilmente el valor buscado. Obtener las unidades es ahora sencillo, restando al número entero tantas decenas como tuviera. Para la parte decimal, se tiene en cuenta que únicamente tendrá 4 valores posibles según se presenta en la tabla siguiente.

2 bits menos significativos	Valor decimal
00	.00
01	.25
10	.50
11	.75

Tabla 5.1 Valores decimales de la temperatura

A la vez que se identifica un dígito, la función lo envía a través del USART y realiza la conversión a *bitmap* para almacenar la información en el *array* correspondiente al nodo indicado.

Los números a representar tendrán una altura de 6 pixeles (cada línea de la pantalla cuenta con 8, pero se dejan 2 como guarda entre líneas) y un ancho de 4. Se usa una representación similar a la representación con 7 segmentos, utilizada en un gran número de relojes digitales, pero cambiando los segmentos por pixeles representados por bits (cada bit es un pixel: 1 encendido, 0 apagado). Por lo tanto, cada dígito hace uso de 4 bytes para ser representado en el *display*. De forma ilustrativa, la conversión se explica de forma gráfica en la Figura 5.5.

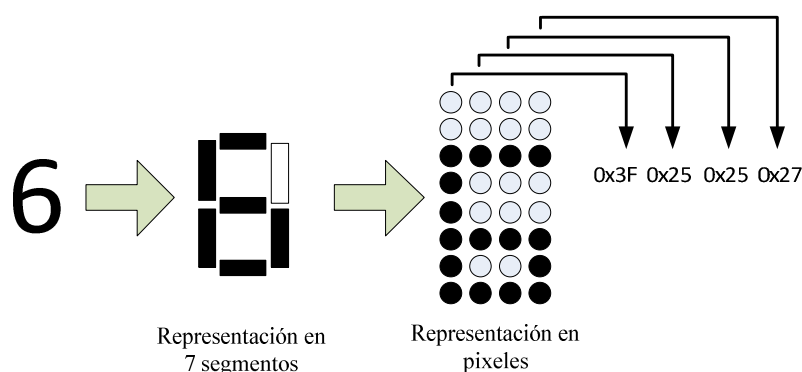


Figura 5.5 Ejemplo de conversión de dígito a bitmap representable en el LCD

### 5.3.3 Nodo satélite RFD

Para esta demostración se han implementado dos nodos satélites que incorporan un microprocesador PIC18F2525, un transceptor MRF24J40MA y un sensor de temperatura LM71. Cada nodo RFD debe comenzar a ejecutar su aplicación una vez que el coordinador ha iniciado la red. Pese a tener una estructura general parecida al coordinador, la funcionalidad de la aplicación de un nodo RFD es bastante más limitada. Por un lado no implementa todas las funciones y posibilidades de MiWi, si no que solamente utiliza las características básicas de conexión, por otro lado, no implementa las funciones de interfaz gráfica con el usuario directamente.

Después de iniciar el procesador, el transceptor y asignar los pines del sensor, el nodo busca una red a la que conectarse, encendiendo un LED de aviso indicando que tal suceso ha sido exitoso. Realizado este proceso, el nodo se mantiene inactivo esperando a recibir un mensaje del coordinador. Cuando se recibe un mensaje, un nuevo LED se enciende para mostrar que el paquete se ha recibido correctamente. Una vez procesado el paquete, se decide si es una petición de nueva temperatura. En caso de ser así, se ejecuta la función encargada de recoger una nueva medición, tras lo cual se envía un mensaje de vuelta al coordinador con la información. Realizada la gestión del paquete recibido, este se descarta y se apaga el LED indicativo. En la Figura 5.6 se muestra el diagrama de flujo de la aplicación que ejecutan estos nodos.

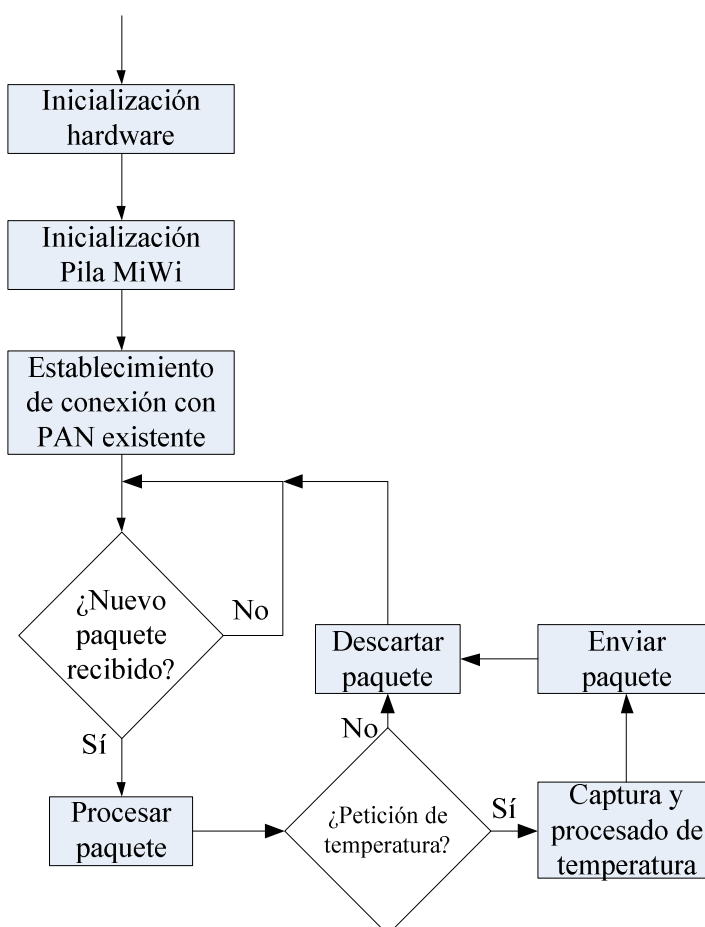


Figura 5.6 Diagrama de flujo de la aplicación del Nodo 2/3



## 5.4 Conclusiones

Con el ejemplo aquí presentado y disponible en el anexo electrónico adjunto a esta memoria, se ha demostrado la utilización todos y cada uno de los elementos descritos en los capítulos 3 y 4.

Se han probado los transceptores y las comunicaciones mediante el protocolo MiWi P2P entre 3 nodos de forma satisfactoria, utilizando una topología en estrella e intercambiando mensajes entre el coordinador y los nodos satélites de forma bidireccional. El *display* y la pantalla táctil han proporcionado una interfaz intuitiva y sencilla de manejar para cualquier usuario. Las comunicaciones serie proporcionan un medio de comunicación cableado con un dispositivo, como un PC, gracias al cual podemos enviar información de control (iniciación de dispositivos y comunicaciones, envío/recepción de mensajes) sin saturar el display. Por último la incorporación a los nodos satélite de los sensores de temperatura otorga a la demostración de un valor añadido, mostrando una utilidad de esta tecnología en el mercado.



# Capítulo 6:

## Presupuesto

El presupuesto de este proyecto está formado por dos tipos de costes más el porcentaje de beneficios. Por un lado se cuentan los costes directos, compuesto por el coste del personal implicado en la finalización del mismo y los gastos en equipo, material y software. Sobre estos costes se calculan los gastos indirectos compuesto por el coste indirecto (electricidad, conexión a internet, amortizaciones, etc.) y el riesgo del proyecto.

### *Costes de personal*

- Personal utilizado: 1.
- Salario neto por persona y mes: 1200€.
- IRPF (20% sobre salario neto): 240€.
- Seguridad social (40% sobre salario neto): 576€.
- Salario bruto por persona y mes: 2016€.
- Meses trabajados: 3 meses.
- Salario bruto total: 6048€.

### *Costes de material<sup>3</sup>*

- MPLAB IDE: gratuito.
- MPLAB C18: 361.47€.
- PicKit 2: 25.53€.
- 2 PIC18F2525: 6.33€.

---

<sup>3</sup> La moneda original de algunos materiales es el dólar. Aquí se presenta una conversión aproximada.

- 2 MRF24J40MA: 14.54€.
- LM71: 0.90€.
- Pantalla táctil Nintendo 5€.
- Pantalla LCD: 6€
- Controlador ADM3202: 2.68€.
- Conector DB9: 2.24€.
- Regulador de tensión LM317T: 1.26€.
- 3 Placas *Protoboard*: 21€.
- Componentes (cableado, resistencias, condensadores, etc.)<sup>4</sup>: 5€.
- 2 Placas de circuito impreso: 12€

### ***Costes indirectos***

- Gastos indirectos: 15% de los costes directos.
- Riesgos: el mayor riesgo del proyecto es el personal, por lo que el sobre coste por riesgo es del 20% de los costes de personal.

### ***Beneficios***

- Calculados como el 25% del coste total.

<b>Descripción</b>	<b>Coste (€)</b>
<b>Costes</b>	8.698,34
Costes directos	6.511,95
Costes de personal	6.048,00
Costes de material	463,95
Costes indirectos	2.186,39
Gastos indirectos	976,79
Riesgo	1.209,60
<b>Beneficios</b>	2.174,59
<b>COSTE TOTAL PROYECTO</b>	10.872,93

*Tabla 6.1 Presupuesto del proyecto*

<sup>4</sup> Debido al bajo valor individual de estos elementos, se ha realizado una aproximación para calcular el valor conjunto. Los componentes específicos pueden consultarse en los esquemas del Anexo C.

## **Capítulo 7:**

# **Conclusiones finales y trabajos futuros**

En este Capítulo se exponen las conclusiones sacadas a la vista de los resultados obtenidos. Posteriormente, se incluye un apartado sobre las posibles ampliaciones del proyecto y otro tipo de trabajos futuros relacionados con el mismo.

### **7.1 Conclusiones finales**

Acabada la placa de desarrollo, los nodos RFD para las pruebas y las mismas pruebas, se puede decir que se han cumplido todos los objetivos planteados en el capítulo primero.

Se ha logrado diseñar una placa de desarrollo para su uso en las distintas asignaturas del departamento de Tecnología Electrónica basadas en microprocesadores. La placa incluye la capacidad de realizar comunicaciones inalámbricas a 2.4GHz y se ha implementado el protocolo propietario MiWi P2P al software de demostración

De forma adicional, la placa desarrollada tiene accesibles tanto los pines no utilizados del microprocesador como otros que, a pesar de que están utilizados, pueden compartirse con otros dispositivos. Esta flexibilidad permite añadir nuevos dispositivos para ampliar la funcionalidad de la placa de desarrollo.

En conjunto, el sistema aquí presentado, tanto el hardware como el ejemplo de aplicación desarrollado para comprobar su funcionamiento, ofrece una multitud de posibilidades a la hora de redactar nuevas prácticas para los alumnos, ya sea partiendo

únicamente desde los esquemas eléctricos y las librerías de Microchip, como usando íntegramente el sistema complejo objeto del proyecto.

También se ha cumplido el objetivo parcial de desarrollar el software de forma modular, separando en librerías distintas las funciones de configuración y uso de los diferentes dispositivos de la placa.

## 7.2 Trabajos futuros

El siguiente paso más intuitivo a la realización de este proyecto, es diseñar unas prácticas propiamente dichas basándose en los trabajos realizados.

Siguiendo una línea continuista, se puede seguir trabajando sobre el software de ejemplo explicado en el capítulo 5 y adjunto en el anexo electrónico, especialmente en la funcionalidad, complejidad y extensión de la aplicación. Al tratarse de una demostración, la aplicación cumple los requisitos básicos para controlar y hacer funcionar todos los dispositivos presentes, por lo que hay amplias posibilidades y caminos a seguir para otorgar de mayor profundidad a la misma. Por ejemplo, se puede evolucionar el sistema de menús presentado en el *display* para permitir al usuario aumentar o disminuir la potencia de transmisión, el canal de operación de la PAN, etc.

De manera paralela y usando el mismo hardware, se puede trabajar con otros protocolos de comunicación inalámbrica, que trabajen en la banda ISM, con distintas características. En este punto, se podría usar Zigbee para aplicaciones que requieran el uso de grandes redes y requieran de mayor control sobre las opciones de transmisión y compatibilidad con otros estándares.

También sería importante estudiar la posibilidad de realizar un prototipo basado en el actual, pero que utilice un microprocesador con un mayor número de pines. Y es que esta es la gran limitación que sufre el PIC18F2525. El microprocesador PIC18F4620 [7] sería un candidato potencial a esta conversión. Este modelo de procesador posee prácticamente las mismas características que el procesador usado, pero cuenta con 40 pines en total (44 según el modelo), permitiendo al usuario conectar un mayor número de dispositivos, ampliando las posibilidades futuras de desarrollo en torno a un mismo esquema.

## **Anexo A:**

# **Instalación de MPLAB IDE y MPLAB C Compiler for PIC18**

En este apéndice se explica cómo descargar e instalar el software utilizado para el desarrollo y prueba del código fuente del proyecto.

MPLAB IDE es el entorno de desarrollo en el que está guardado el proyecto. Dispone de un editor de texto en C, así como funcionalidad para simular y depurar el código ejecutándolo directamente sobre el microprocesador (para ello es necesario disponer de una conexión entre éste y entorno de Microchip Inc.

Este software puede ser descargado en su última versión desde el siguiente enlace:

[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en019469&part=SW007002](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002)

Una vez descargado, se descomprime el archivo comprimido y se arranca el ejecutable *Setup.exe*. Durante el proceso de instalación basta con seguir los pasos que se indican en los cuadros de dialogo para completar con éxito el proceso.

Por otro lado, también es necesario descargar e instalar el compilador de C específico para el entorno de desarrollo y la familia de PIC elegida (en este caso, para PIC18). El compilador puede descargarse en

[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en010014](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014)

En el enlace superior se puede acceder a la versión reducida del programa, pero para ejecutar correctamente la Pila MiWi proporcionada por Microchip Inc. es necesaria la versión completa con licencia válida (ya sea en periodo de prueba o una licencia

actualizada). Además, también se hace necesario que el compilador sea una versión igual o superior a la v3.04 (este proyecto ha sido desarrollado utilizando la versión v3.37.01).

Una vez descargado el ejecutable, se hace doble click sobre él y se siguen las indicaciones del cuadro de instalación.

Para más información acerca de MPLAB IDE o MPLAB C Compiler for PIC18 se pueden consultar las guías de usuario disponible en los mismos enlaces indicados para la descarga del software.



## Anexo B:

# Configuración de Proyecto nuevo en MPAB IDE

Debido a problemas con el compilador, es posible que haya que crear un nuevo proyecto si se desea ejecutar el código en una máquina distinta a la que el proyecto original (adjunto a esta memoria) fue creado y compilado.

Una vez abierto el entorno de desarrollo, se crea un nuevo proyecto mediante *Project > Project Wizard* donde se elige el microprocesador (PIC18F2525), el compilador (MPLAB C18), el nombre y ruta del proyecto, y los archivos con código que lo componen. Es recomendable mantener la misma estructura de archivos y carpetas que en el proyecto original para evitar confusiones y/o errores de compilación al no encontrar los archivos requeridos.

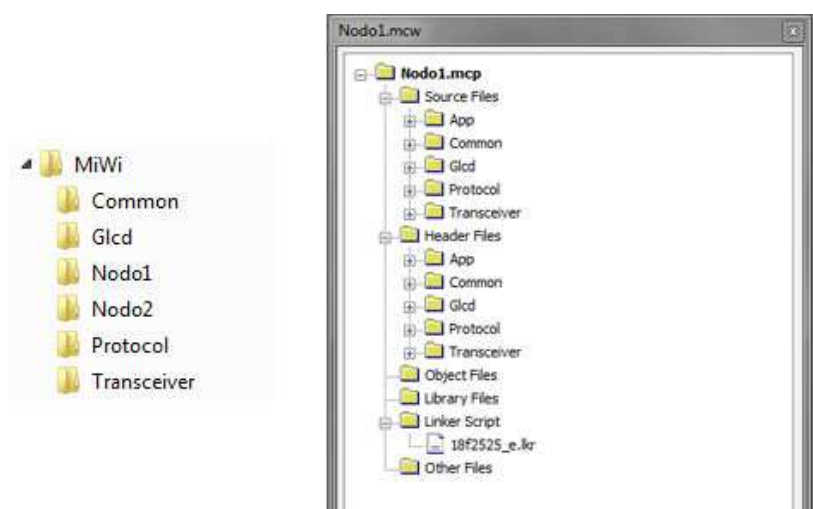


Figura B.1 Estructura de Archivos en Windows (izq.) y en MPLAB IDE (dcha.)

A continuación, es necesario modificar las opciones del proyecto en *Project > Build Option > Project* tal y como se muestra en la Figura B.2.

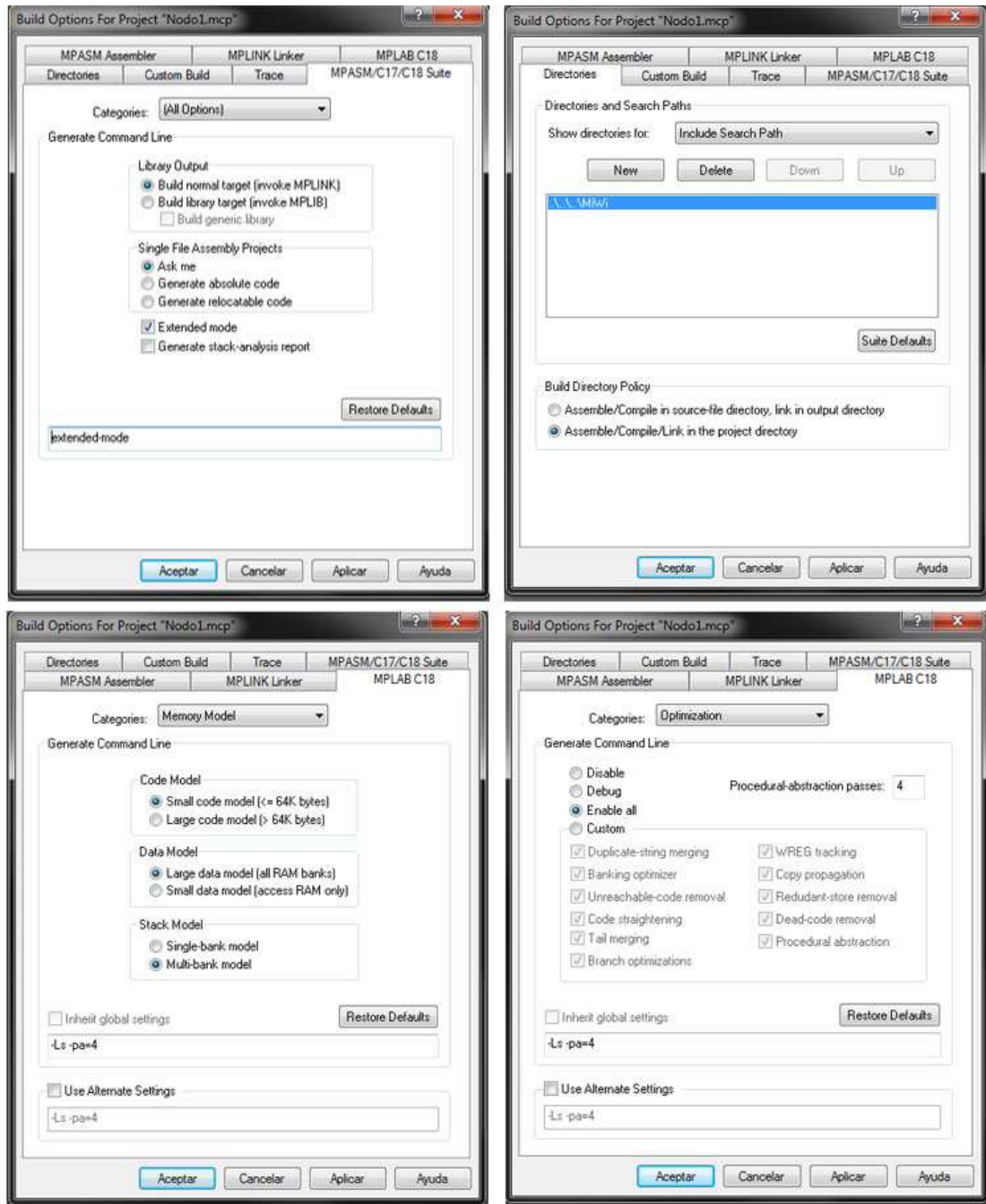


Figura B.2 Configuración de las Opciones de Proyecto

Cabe destacar que, en la captura superior derecha de la Figura B.2, la ruta de búsqueda de archivos, *Include Search Path*, puede variar según el lugar donde se haya decidido guardar el proyecto con todos sus archivos.

El proceso aquí descrito crea un único proyecto que ejecutará el código de uno de los nodos, dependiendo de los archivos de código que se cargue en él. Del mismo modo, hay que crear un segundo proyecto para poder compilar y ejecutar el otro nodo.



## **Anexo C:**

# **Esquema eléctrico del Hardware**

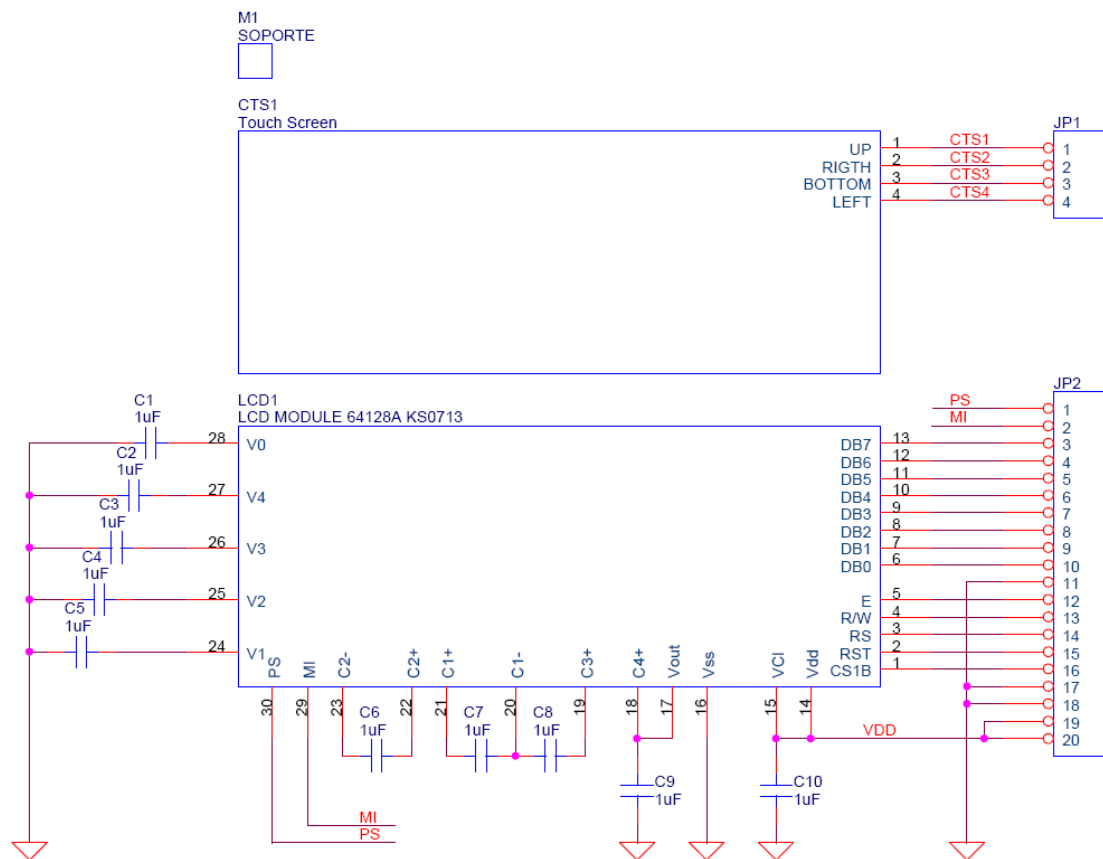


Figura C.1 Esquema eléctrico de las pantallas LCD y táctil

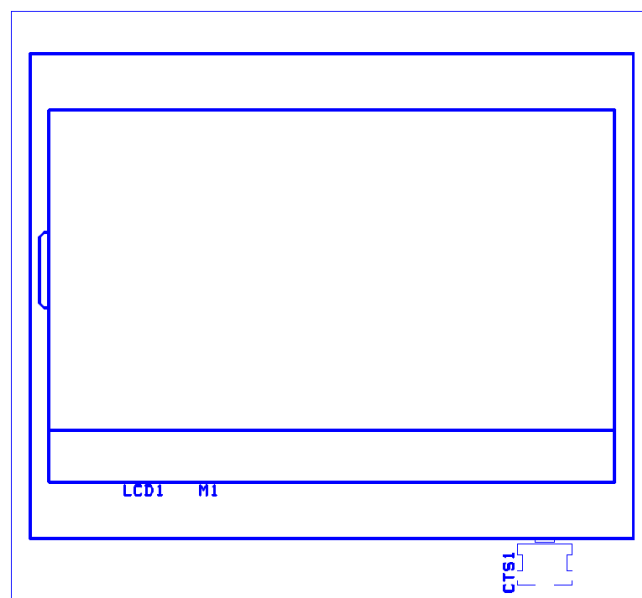


Figura C.2 Esquema SST de las pantallas LCD y táctil

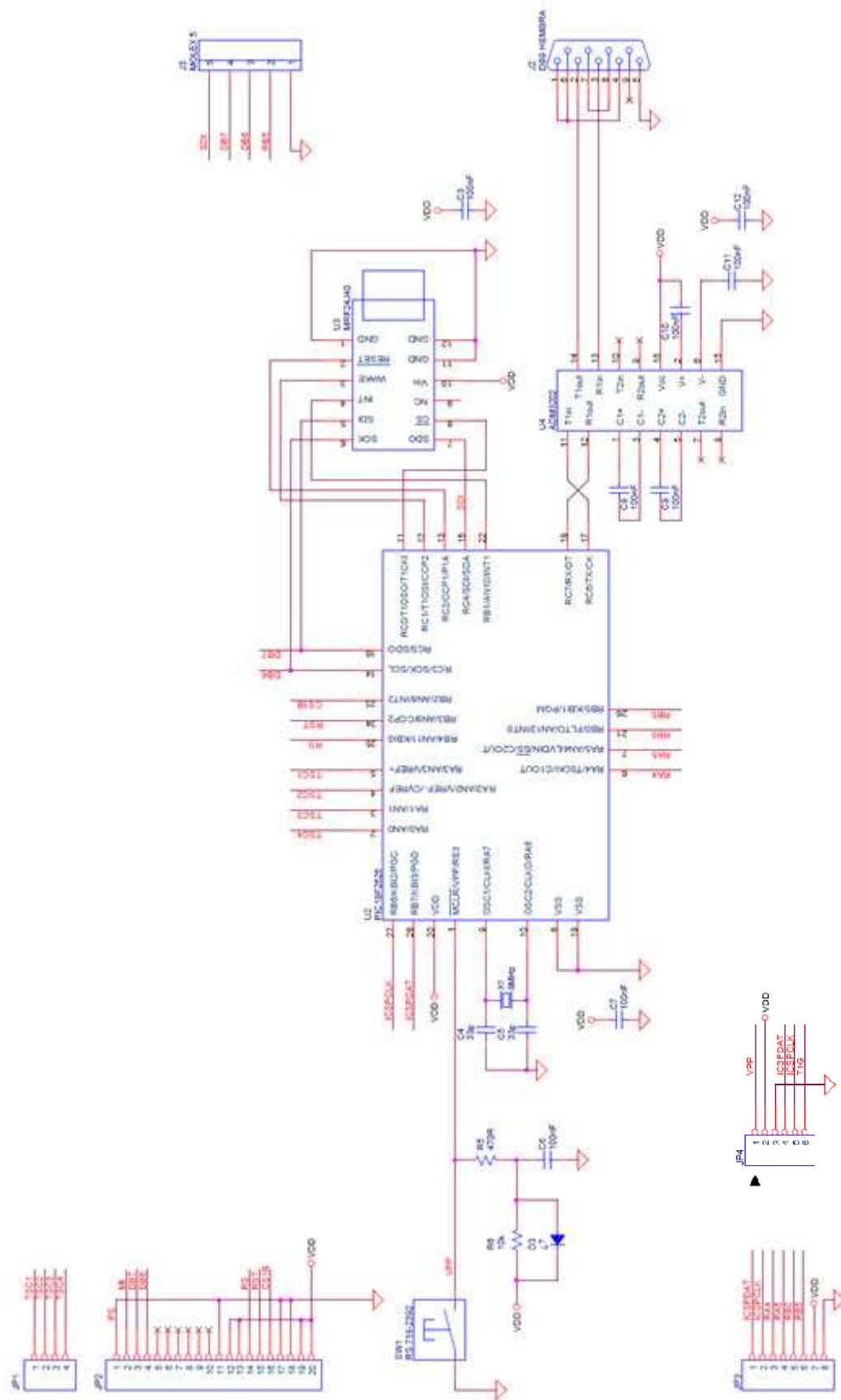


Figura C.3 Esquema eléctrico de la CPU

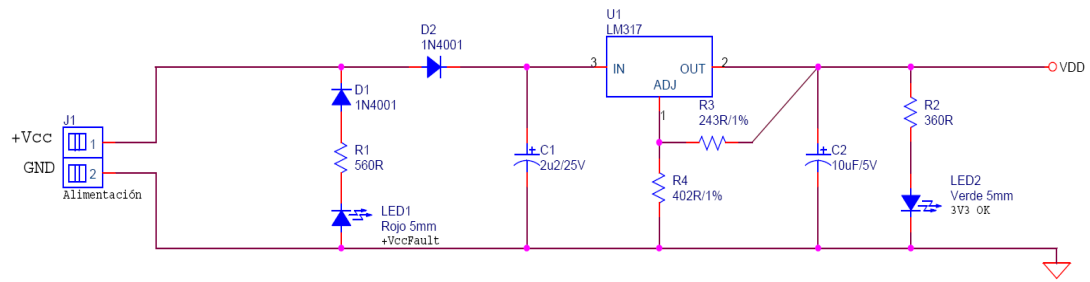


Figura C.4 Esquema eléctrico del regulador de tensión

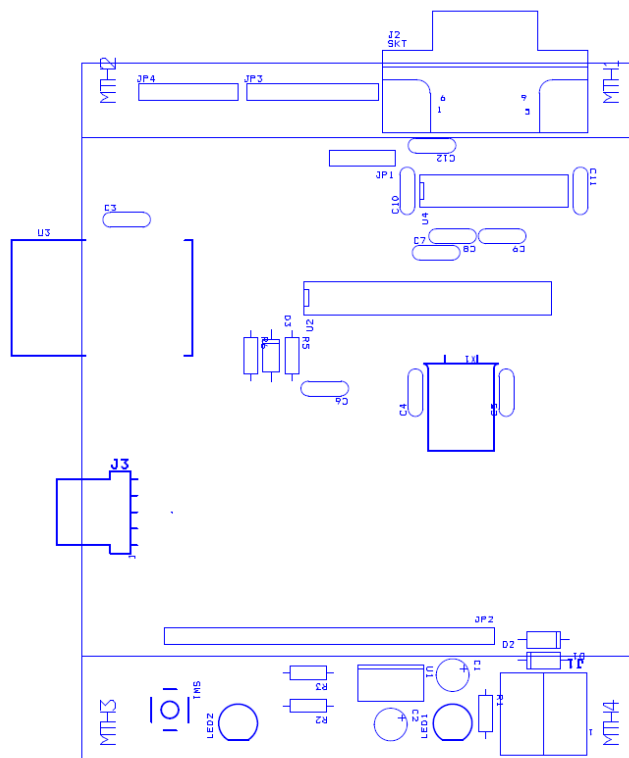


Figura C.5 Esquema SST de la CPU y el regulador



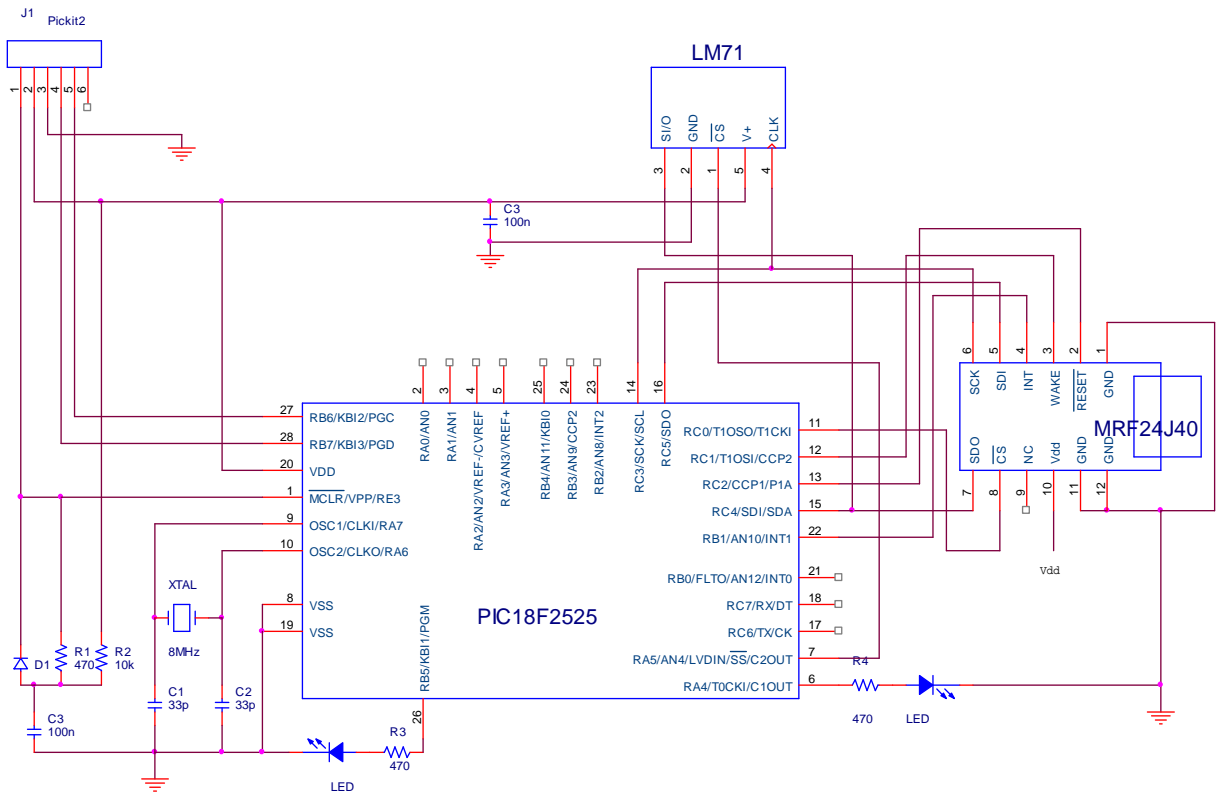


Figura C.6 Esquema eléctrico del nodo RFD utilizado en las pruebas



# Glosario

ACK	Acknowledgement
AFH	Adaptative Frequency Hopping
AIS	Adaptative Interference Supression
APL	Application Layer
APS	Adaptative Packet Selection
APSch	Adaptative Packet Scheduling
AWMA	Alternating Wireless Medium Access
BPSK	Binary Phase Shift Keying
BW	BandWidth
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCM	Counter with CBC-MAC
CSMA/CA	Carrier sense multiple access with collision avoidance
CTR	Counter Mode
DEV	Device
DLL	Data Link Layer
ECB	Electronic CodeBook
ED	Energy Detect
FFD	Full Function Device
GFSK	Gaussian frequency-shift keying

GPIO	General Purpose Input/Output
IEEE	Institute of Electrical and Electronics Engineers
IFS	Inter Frame Spacing
ISM	Industry, Science and Medical
LLC	Logical Link Control
LQI	Link Quality Indication
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Medium Control Access
MCU	Microcontroller Unit
MiWi	Microchip Wireless
NWK	Network Layer
O-QPSK	Offset Quadrature Phase Shift Keying
PAN	Personal Area Network
PHR	PHY Header
PHY	Physical
PNC	PicoNet Coordinator
PPDU	PHY Protocol Data Unit
PSDU	PHY Service Data Unit
PTA	Packet Traffic Arbitration
QoS	Quality of Service.
RFD	Reduced Function Device
SHR	Synchronization Header
TDD	Time Division Duplex
TEA	Tiny Encryption Algorithm
UMTS	Universal Mobile Telecommunications System
WBAN	Wireless Body Area Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
XTEA	eXtended Tiny Encryption Algorithm

# Referencias

- [1] A. Papacetzzi, F. Martín. "Wireless Personal Area Network (WPANs) & Home Networking." Escuela de Ingeniería, Universidad de las Américas Puebla. Departamento de Ingeniería Electrónica (2003).
- [2] J. M. Eugenio Barroso, S. Camacho Domínguez. "El estándar Bluetooth". [http://www.investigalog.com/ingenierias\\_y\\_sistemas\\_de\\_gestion/el-estandar-bluetooth/](http://www.investigalog.com/ingenierias_y_sistemas_de_gestion/el-estandar-bluetooth/) [Accedido a 18/09/2011]
- [3] IEEE 802 LAN/MAN Standards Committee. <http://www.ieee802.org/15/> [Accedido a 18/09/2011]
- [4] IEEE 802.115.2. <http://standards.ieee.org/getieee802/download/802.15.2-2003.pdf> [Accedido a 18/09/2011]
- [5] El estándar IEEE 802.15.4. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lem/archundia\\_p\\_fm/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf) [Accedido en 18/09/2011]
- [6] E. T. Won, D. Shin, D.K. Jung, Y.J. Oh, T.Bae, H. Kwon, C. Cho, J. Son, D.O'Brien, T. Kang, T. Matsumura. "Visible Light Communication : Tutorial". [http://ieee802.org/802\\_tutorials/2008-03/15-08-0114-02-0000-VLC\\_Tutorial\\_MCO\\_Samsung-VLCC-Oxford\\_2008-03-17.pdf](http://ieee802.org/802_tutorials/2008-03/15-08-0114-02-0000-VLC_Tutorial_MCO_Samsung-VLCC-Oxford_2008-03-17.pdf) [Accedido a 05/10/2011]
- [7] Microchip, 2004. PIC18F2525 Datasheet. USA
- [8] Microchip, 2010. MRF24J40MA Datasheet. USA
- [9] Microchip, 2010. MRF24J40 Datasheet. USA
- [10] Displaytech, 2005. LCD MODULE 64128A COG SERIES Datasheet, USA
- [11] LCD Driver IC Team, 2000. Controlador KS0713 Datasheet

- [12] Displaytech, 2008. LCD MODULE 64128M COG SERIES Datasheet, USA
- [13] Sitronix 2007. Controlador ST7565R Datasheet. USA
- [14] HanTouchUSA. "How it work: 4-wire Analog-Resistive TouchScreens".  
<http://www.sparkfun.com/datasheets/LCD/HOW%20DOES%20IT%20WORK.pdf>  
df [Accedido a 07/06/2011]
- [15] Analog Device, 2006. ADM3202 Datasheet. USA
- [16] National Semiconductor, 2005. LM317 Datasheet. USA
- [17] MiWi DE.  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2113&param=en520414](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2113&param=en520414) [Accedido 02/03/2011]
- [18] The Alliance. <http://www.zigbee.org/About/AboutAlliance/TheAlliance.aspx>  
[Accedido a 06/10/2011]
- [19] V. Moretto. "Wireless (IEEE 802.15.4)".  
[http://www.tecnoimprese.it/user/File/Eventi/RF08-14feb\\_melchioni.pdf](http://www.tecnoimprese.it/user/File/Eventi/RF08-14feb_melchioni.pdf)  
[Accedido en 25/09/2011]
- [20] AN1204 "MiWi P2P Wireless Protocol".
- [21] AN1283 "Microchip Wireless (MiWi) Media Access Controller".
- [22] Cifrado de Feistel. [http://es.wikipedia.org/wiki/Red\\_de\\_Feistel](http://es.wikipedia.org/wiki/Red_de_Feistel) [Accedido a 19/10/2011]
- [23] AN1284. "MiWi Application Programming Interface"
- [24] National Semiconductor, 2005. LM71 Datasheet. USA